

Profiling Computational Thinking and Multimedia Pre-Training Integration in Programming Instruction: Lecturers' Perspectives

Mimi Zairul Mohmad Fuzi¹, Siti Nazleen Abdul Rabu^{1*},
Wan Ahmad Jaafar Wan Yahaya¹

¹ Centre for Instructional Technology and Multimedia, Universiti Sains Malaysia, Penang, Malaysia

*Corresponding Author: snazleen@usm.my

Received: 10 January 2026 | Accepted: 25 March 2026 | Published: 1 April 2026

DOI: <https://doi.org/10.55057/ijares.2026.8.2.24>

Abstract: *This study examines lecturers' perspectives on the integration of Computational Thinking (CT) and Multimedia Pre-Training in teaching programming at matriculation colleges. Using qualitative methods with in-depth interviews with five lecturers, this study identifies teaching challenges, existing strategies, perceptions of CT, and the role of multimedia pre-training in supporting student learning. Findings indicate that students face difficulties in understanding basic concepts and low levels of engagement, while existing teaching strategies such as step-by-step demonstrations, practical exercises, and the use of visual aids can help overcome these challenges. Lecturers view CT as an effective tool for improving problem-solving skills, and multimedia pre-training improves students' initial understanding and engagement. This study contributes to the development of an integrated pedagogical model that can improve the effectiveness of teaching programming at the matriculation level.*

Keywords: Computational Thinking, Multimedia Pre-Training, Programming Instruction, Matriculation Colleges, Pedagogical Strategies

1. Introduction

As educational systems increasingly emphasise digital, programming and computational thinking (CT) skills have become a basic requirement in science and technology education as students must not only learn how to write code but also develop problem-solving and analytical skills essential for future work and study (Perez Valdes et al., 2025; Ubaidullah et al., 2023). Programming not only serves as the foundation for the development of modern technology, but also plays an important role in shaping a systematic, logical and structured way of thinking among students (Sibia et al., 2025). In this regard, lecturers at pre-university educational institutions such as matriculation colleges play a very important role in ensuring that students can master the concept of programming effectively before moving on to higher levels of study as their pedagogical decisions influence both students' conceptual understanding and engagement (Ubaidullah et al., 2023).

However, from the lecturers' perspective, teaching programming especially to matriculation students still faces various challenges. Among the main issues frequently raised is the low level of student achievement in solving programming problems, even when the syllabus has been fully delivered. Students have difficulty understanding fundamental programming concepts

such as control structures and logical reasoning, and struggle to apply problem-solving strategies due to cognitive overload and limited experience with abstract concepts (Omeh et al., 2025). This situation reflects a gap between teaching methods and students' cognitive abilities and learning styles, particularly in introductory programming where the complex interplay of syntax, logic, and problem decomposition can overwhelm novice learners (Gao et al., 2023).

In addition, constraints related to limited teaching time and students' diverse academic backgrounds compound lecturers' difficulties. Within a single class, some students may have a basic programming foundation while others have no prior exposure at all, making it challenging to plan appropriate strategies for all learners simultaneously. Lecturer-centered teaching approaches have been found to be less effective in addressing the differentiated needs of students, particularly in supporting weaker students to build meaningful understanding and problem-solving skills. Recent research highlights that time constraints for planning and instruction, combined with limited pedagogical and technological resources, remain persistent barriers to adapting teaching to diverse learner profiles in programming education (Fagerlund & Rissanen, 2024).

In terms of teaching materials, lecturers often report a lack of interactive and appropriate resources to support the teaching of programming. Most existing materials are static notes and simple code examples, which do not adequately help students build mental models of abstract concepts. This increases students' cognitive load, making it harder to process complex information, and places additional pressure on lecturers to achieve desired learning outcomes within limited time (Mayer et al., 2024).

In this context, the integration of computational thinking and multimedia pre-training principles is seen as an approach with the potential to address these challenges. CT emphasises problem decomposition, pattern recognition, abstraction, and algorithm construction skills that can help students tackle programming problems more systematically (Perez Valdes et al., 2025; Ubaidullah et al., 2023). Meanwhile, the Cognitive Theory of Multimedia Learning (CTML) suggests that pre-training learners with key concepts using multimedia can reduce extraneous cognitive load and facilitate deeper processing when formal instruction begins (Mayer et al., 2023).

Nevertheless, implementing this integrated approach is not without challenges. Designing CT-based activities that are suitable for students' current skill levels, lecturers' technological proficiency, and the feasibility of using learning technologies in real classroom contexts are persistent concerns. Additionally, lecturers question whether these strategies improve student engagement or simply add to their existing instructional workload (Perez Valdes et al., 2025).

Thus, the main problem addressed in this study is the lack of in-depth understanding of teaching challenges, pedagogical considerations, and lecturers' perceptions of integrating CT and multimedia pre-training principles in programming instruction at matriculation colleges. While many prior studies have focused on student outcomes, fewer investigations have explored lecturers' perspectives, even though they are the primary implementers of teaching approaches.

Therefore, this study was conducted to qualitatively explore lecturers' perspectives to understand how they see the potential and constraints of integrating computational thinking and multimedia pre-training in teaching programming. Specifically, the objectives of this study are:

1. to identify the teaching challenges faced by lecturers in teaching programming in matriculation colleges;
2. to explore lecturers' perceptions of the use of computational thinking as a pedagogical strategy in teaching programming problem-solving; and
3. to examine lecturers' views on the role of multimedia pre-training principles in supporting the teaching process and increasing student engagement and understanding.

The findings of this study are expected to provide a more comprehensive picture of the reality of teaching programming from the lecturers' perspective, as well as contribute to the development of teaching approaches and learning application designs that are more practical, effective and in line with the actual needs in matriculation classrooms.

Significance and Contribution

This study has significant significance and contribution to various parties, including the government, the education community, industry, and the development of knowledge in the field of programming education. By examining lecturers' perspectives on the integration of computational thinking and multimedia pre-training principles in programming teaching, this study provides a more comprehensive understanding of the real challenges of implementing innovative pedagogical approaches at the matriculation college level (Liu et al., 2024).

From the government's perspective, the findings of this study can support the planning and improvement of educational policies, especially in the teaching of Computer Science and Information Technology at the pre-university level. Information on teaching challenges, time constraints, and pedagogical support needs faced by lecturers can be used as a reference in curriculum development, the provision of professional training, and the formulation of policies related to the integration of technology and computational thinking in the national education system (Abu Bakar et al., 2025)

In terms of the education community, this study directly benefits lecturers and institutions of learning by providing empirical guidance on how computational thinking and multimedia pre-training can be used as more effective teaching strategies. The findings of this study can help lecturers improve teaching practices, increase student engagement, and reduce students' difficulties in understanding abstract programming concepts.

For the industry, this study contributes to the development of human capital that is better prepared to face the needs of the technology and digital sectors. Graduates exposed to computational thinking-based teaching are expected to have more systematic problem-solving skills that are relevant to the needs of the information technology industry, thus increasing the marketability of the workforce.

From a new knowledge perspective, this study enriches the existing literature by presenting a lecturer's perspective that is still underexplored in research related to programming teaching. This study also contributes to strengthening the theory of Cognitive Multimedia Learning by relating the principles of multimedia pre-training to actual pedagogical practices, thus opening up space for further research in instructional design and learning applications based on computational thinking.

Scope of the research

This study focuses on matriculation lecturers who teach programming to students in the Four Semester System in the Science department. The scope of the study focuses on the lecturers' experiences and perceptions of teaching challenges as well as the integration of computational thinking and multimedia pre-training principles in teaching programming problem solving. This study uses a qualitative approach and does not assess student achievement quantitatively. Therefore, the study findings are limited to the context of teaching in matriculation colleges for the Four Semester System and are not intended to be generalized to other educational levels.

2. Literature Review

This literature review aims to provide a theoretical and empirical basis to understand lecturers' perspectives on the integration of Computational Thinking (CT) and Multimedia Pre-Training in teaching programming at matriculation colleges. The main focus of the literature review is to identify the challenges of teaching programming, evaluate the effectiveness of CT and multimedia-based strategies, and survey lecturers' perceptions of the use of innovative learning applications that support students' problem-solving skills.

Previous studies have shown that teaching programming at the pre-university level faces various challenges, including students' non-uniform prior knowledge levels, difficulty understanding abstract concepts, and a lack of interactive teaching resources (Sala Hudin, 2023). Lecturers often report that students have difficulty applying logical concepts to solve real-world problems, while limited teaching time makes it difficult to implement more student-centered activities. This highlights the need for teaching strategies that not only focus on content delivery, but also build critical thinking and problem-solving skills among students.

In this context, Computational Thinking (CT) has been identified as an effective approach to improve problem-solving skills. CT includes several important elements such as decomposition, pattern recognition, abstraction, and algorithmic thinking that allow students to break complex problems into smaller parts and develop solution logic systematically. Previous studies have shown that the integration of CT in STEM teaching, especially programming, can improve students' ability to solve problems effectively (Chang, Liang, & Tsai, 2024). However, lecturers' perceptions of CT still vary; some lecturers consider it relevant and practical, while others see challenges in terms of time and student readiness to follow this approach.

In addition, the principle of Multimedia Pre-Training according to the CTML emphasizes the importance of providing students with initial knowledge through static or dynamic visuals before introducing more complex concepts. Pre-training aims to build basic knowledge schemas, increase engagement, and support long-term information retention (Mayer, 2022). Studies have shown that the use of pre-training can improve students' understanding of programming concepts and reduce cognitive load when solving problems. However, the implementation of pre-training depends on the lecturer's perception and experience in designing appropriate teaching materials and integrating learning applications.

Recent studies also examine the integration of CT and multimedia in digital applications, highlighting potential benefits such as support for problem-based learning, increased interactivity, and improved problem-solving outcomes (Chen & Huang, 2024). CT and multimedia-based applications have been proven to support problem-based learning, enhance interactivity, and build students' problem-solving skills. However, practical challenges such as

development costs, technological accessibility, and students' ability to use the applications effectively are often issues for lecturers. Lecturers' perceptions of the effectiveness of these applications play a crucial role in the successful implementation in the classroom.

The literature review also highlights that lecturers' perspectives on innovative teaching strategies are influenced by several factors, including teaching experience, technological readiness, institutional support, and the effectiveness of the strategies in real-world contexts. The absence of studies that specifically focus on the views of matriculation-level lecturers on the integration of CT and pre-training in teaching programming indicates a gap in the literature that needs to be filled. Existing studies focus more on student effectiveness or technical analysis of applications, while lecturers' perspectives are still underexplored, especially in the pre-university context in Malaysia.

Overall, the literature reviewed shows that CT and multimedia pre-training offer great potential to improve the effectiveness of programming teaching. However, there are controversies and practical challenges from the lecturer perspective that require further study. The absence of studies that examine lecturer perceptions in the context of matriculation emphasizes the importance of this study to fill the literature gap, thereby providing practical guidance for improving programming teaching strategies that are technology-based and focused on student problem-solving.

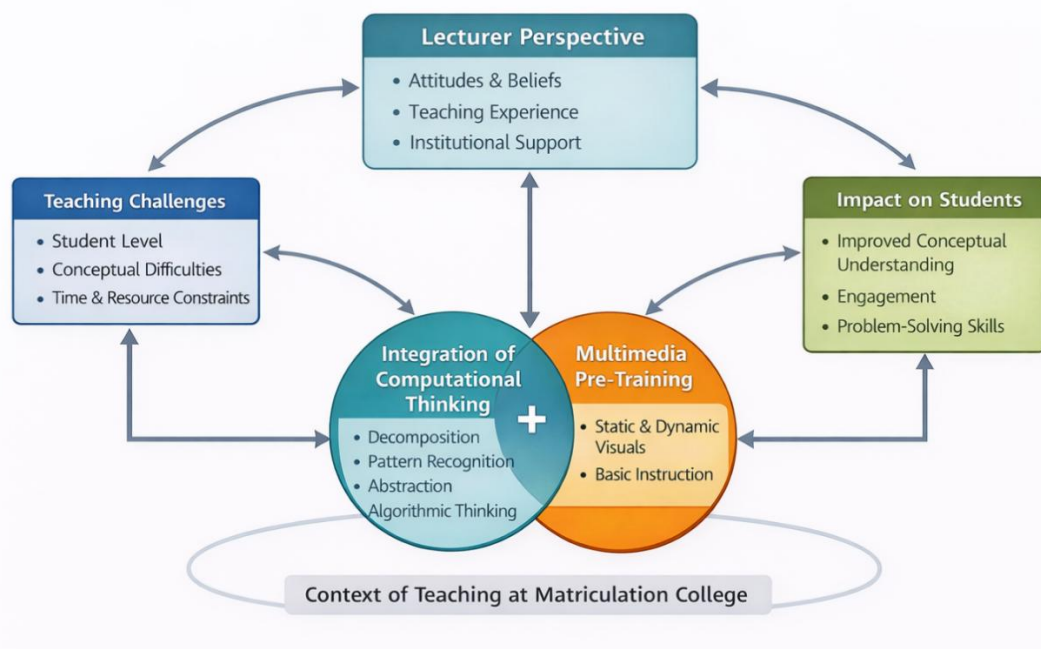


Figure 1: Conceptual Framework

This conceptual framework diagram illustrates the relationship between the main themes of the study in the context of teaching programming in matriculation colleges. The two main themes, namely Computational Thinking (CT) Integration and Multimedia Pre-Training, show the teaching strategies that are combined to support student learning. The lecturer's perspective is positioned as an important factor that influences how these two strategies are implemented, including attitudes, experiences, and institutional support. Teaching challenges, such as student knowledge level, conceptual difficulty, and time and resource constraints, are depicted as contexts that influence the implementation of strategies. The interaction between the main themes, lecturer's perspectives, and teaching challenges leads to implications for students,

including increased conceptual understanding, engagement, and problem-solving skills. This diagram emphasizes that this qualitative study focuses on descriptively exploring the lecturer's experiences and views on the context of teaching programming in matriculation colleges.

3. Methodology

This study uses a qualitative research method to explore lecturers' perspectives on the integration of Computational Thinking (CT) and Multimedia Pre-Training in programming teaching at matriculation colleges. A total of five participants were interviewed to obtain in-depth and contextual data. The selection of a qualitative method allows this study to emphasize understanding the experiences, views, and perceptions of lecturers holistically, in line with the study's objectives to identify teaching challenges, perceptions of CT and pre-training, and the effectiveness of learning applications.

The Research Design of this study is Basic Qualitative Inquiry. This design was chosen because it is suitable for basic research that aims to understand phenomena from the participants' perspectives without the need for complex theories or tested hypotheses. Basic Qualitative Inquiry allows for an emphasis on the subjective experiences of participants and allows researchers to collect rich and descriptive data, in line with the study's objectives to explore lecturers' perspectives in depth.

Data Collection was conducted through semi-structured interviews and observations. Interviews were used to obtain participants' views on teaching challenges, their perceptions of CT and pre-training, and their experiences using learning applications. Observations were conducted to assess lecturers' interactions with students and learning applications during actual teaching sessions. The selection of this method was in line with the study objectives because it allowed for contextual and in-depth data collection, and provided participants with the opportunity to share their experiences freely.

The study used purposive sampling, with the target population consisting of lecturers who teach programming at matriculation colleges. This strategy was chosen because participants were selected based on their experience and knowledge relevant to the study topic. A sample size of five people was considered sufficient in a basic qualitative study because it allowed the researcher to focus on in-depth analysis of each participant and ensure that the data obtained was rich and meaningful.

Data Analysis was conducted using manual coding and thematic analysis using Microsoft Word. Interview transcripts and observation data were analyzed to identify themes, patterns, and key categories that emerged from the participants' perspectives. This method was appropriate for the study because it allowed the researcher to systematically structure the data, identify relationships between themes, and maintain the original context of the participants' experiences. The manual approach also gave the researcher full control in assessing the explicit and implicit meanings in the data.

Ethical Considerations were taken into account throughout the research process. Ethical approval was obtained from the institution, and all participants were informed about the objectives of the study, confidentiality of information, and the right to withdraw from the study at any time. Written consent was obtained from all participants before data collection was conducted.

Data Validity and Reliability were ensured through data triangulation, where interviews and observations were combined to obtain a more comprehensive picture. The researcher also conducted repeated checks on the transcripts and themes developed to ensure the accuracy and consistency of the analysis.

Overall, the methodology of this study was designed to ensure systematic, comprehensive, and ethical data collection and analysis. Qualitative methods, Basic Qualitative Inquiry design, data collection techniques, sampling, and thematic analysis were all adapted to achieve the study objective of understanding lecturers' perspectives in depth on the integration of CT and multimedia pre-training in teaching programming.

4. Findings

This study examined lecturers' perspectives on the challenges and strategies of teaching programming in matriculation colleges, as well as their views on the integration of Computational Thinking (CT) and Multimedia Pre-Training. Interviews and observations conducted with five participants yielded rich data, which were categorized into four main themes based on thematic analysis: Teaching Challenges, Existing Teaching Strategies, Perceptions of Computational Thinking, and the Role of Multimedia Pre-Training.

Teaching Challenges was the most prominent main theme. Lecturers reported that one of the main issues was students' difficulty in understanding basic programming concepts. Coding analysis showed that participants often emphasized that although basic concepts seem simple, students sometimes fail to grasp these ideas thoroughly. This in turn affects student engagement in learning activities. Several participants also emphasized that students' uneven level of understanding caused lecturers to spend more time explaining concepts repeatedly. Initial coding indicated that the main issues were related to difficulty in understanding, low engagement, and low information retention. This challenge shows that teaching programming at the matriculation level requires an approach that not only focuses on delivering content but also builds a foundation of logical thinking and problem-solving strategies among students.

For the second theme, Existing Teaching Strategies, the study found that lecturers used a variety of approaches to help students understand programming. The most frequently used strategies included step-by-step demonstrations, the use of visual aids, and practical exercises that allowed students to try out programming code themselves. Initial coding showed that hands-on exercises and collaborative learning were important elements in the strategies used, although their effectiveness varied according to the students' ability level. Several lecturers emphasized that demonstration approaches and practical examples were more effective for students who understood quickly, while students who were less fast needed additional support. Thematic analysis showed that although these strategies were helpful in teaching, there were constraints such as time constraints, large student numbers, and limited technological resources.

The third theme, Perceptions of Computational Thinking (CT), shows that lecturers generally have a positive view of the use of CT as a teaching strategy. Initial coding emphasized that CT helps students to decompose problems, recognize patterns, and build solution logic systematically. Lecturers believe that this approach can improve students' ability to solve programming problems in a structured manner. However, several participants emphasized that challenges exist when students need to apply CT principles independently, especially in decomposing complex problems. Therefore, lecturers need to provide gradual guidance and

ongoing support to ensure that students can use CT effectively. This finding emphasizes that although CT is well-received as a pedagogical concept, its effectiveness depends on the lecturer’s experience, willingness, and guidance approach.

The fourth theme is the Role of Multimedia Pre-Training, which emphasizes the use of visual materials before students encounter complex programming topics. Initial coding showed that the use of dynamic and interactive multimedia was perceived as more engaging and provided students with opportunities to build knowledge schemas before the actual class. However, there were also challenges related to logistical and resource aspects, such as lack of computers, internet connectivity, or students’ discipline in using the applications independently. This analysis shows that while multimedia pre-training offers great potential, its success depends on the resources available and the discipline of the students.

Overall, the study findings indicate that lecturers in matriculation colleges face consistent challenges in teaching programming, particularly with regard to understanding basic concepts and student engagement. Existing teaching strategies such as demonstrations and practical exercises help to alleviate these issues, but their effectiveness depends on the level of students and the support provided. CT integration is seen as a beneficial approach to building problem-solving skills, while multimedia pre-training can increase engagement and early understanding, despite logistical challenges and dependence on student discipline.

These findings suggest that lecturers’ perceptions, their experiences, and the teaching context play a significant role in the effectiveness of innovative teaching strategies. Interview and observation data provide a clear picture that integrated strategies that combine CT and multimedia pre-training have the potential to enhance programming learning, but successful implementation requires institutional support, adequate resources, and structured pedagogical guidance. Initial coding of each theme provides a basis for deeper analysis, which will be explored in the main results and observations, including the relationships between themes, challenges, and the effectiveness of strategies in real classroom contexts.

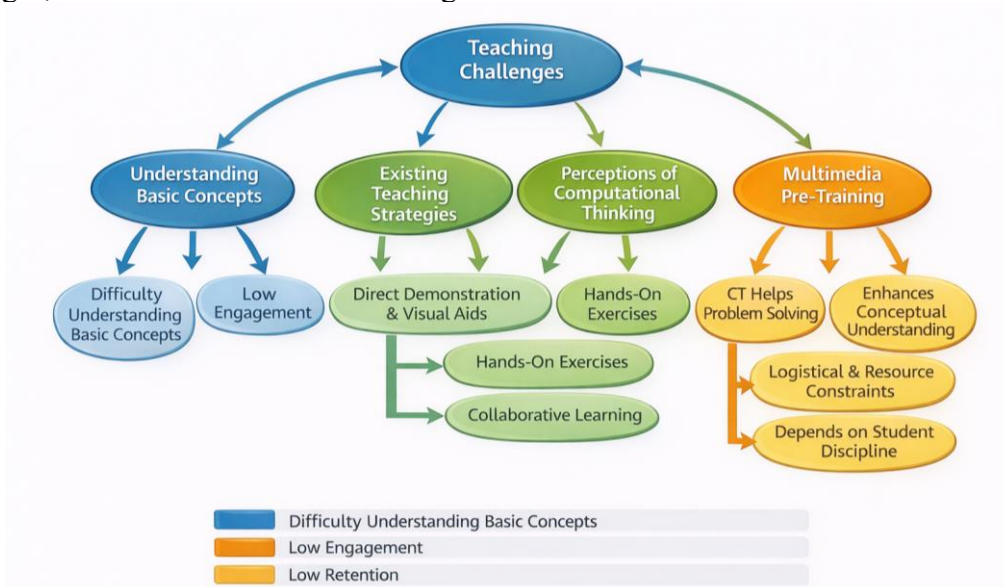


Figure 2: Mind map on programming education challenges

5. Main Results and Observations

The interview and observation analysis in this study showed that lecturers faced various experiences and challenges in teaching programming in matriculation colleges. The initial coding obtained from the participants' responses was manually compiled and analyzed to form main themes that reflect the current pedagogical reality. Four main themes were identified: Teaching Challenges, Existing Teaching Strategies, Perceptions of Computational Thinking (CT), and the Role of Multimedia Pre-Training. These themes were formed from the repetition of participant response patterns, similar codes, and consistent concepts emerging from the interviews.

Teaching Challenges emerged from coding that emphasized students' difficulty in understanding basic concepts, low engagement, and limited retention. Participants reported that although basic concepts seemed simple, students often failed to understand them correctly. Several participants emphasized that the differences in students' ability levels caused lecturers to have to re-explain repeatedly, which increased teaching stress. These codings were combined to form themes because they showed recurring patterns in lecturers' perceptions of the main obstacles in teaching programming, particularly in the context of understanding basic concepts and student engagement.

Existing Teaching Strategies emerged from coding that included the use of step-by-step demonstrations, practical exercises, and visual aids. Participants emphasized that hands-on and collaborative learning approaches were used to enhance student understanding. This theme emerged because participants' coding showed a consistent pattern of teaching techniques used to overcome student challenges. This is consistent with evidence that active learning techniques and visual scaffolds improve comprehension and motivation in programming education (Selamat et al., 2024). While these strategies were helpful, several participants emphasized that effectiveness depended on student level, class size, and resource support, indicating that existing strategies did not always meet all student needs.

Perceptions of Computational Thinking (CT) emerged from coding that emphasized that CT helps students to deconstruct problems, recognize patterns, and construct solution logic. This theme emerged because participants consistently emphasized the benefits of CT in helping students solve programming problems systematically. This finding reflects research demonstrating that CT frameworks support systematic problem-solving and logical reasoning in programming contexts (Balacuit et al., 2024; Ubaidullah et al., 2023). However, the coding also revealed some challenges, for example when students had to apply CT independently. Therefore, this theme not only reflects the lecturer's positive views, but also implementation challenges that require step-by-step guidance.

The Role of Multimedia Pre-Training emerged from coding emphasizing that interactive materials or applications help students build initial conceptual understanding, increase engagement, and support retention. Participants consistently stated that multimedia pre-training had the potential to enhance learning before actual instruction. Research on multimedia learning suggests that pre-training with visual and interactive resources can reduce cognitive load and facilitate schema development (Mayer, 2022). However, coding also highlighted logistical constraints, student discipline issues, and resource dependency as practical barriers, indicating that the effectiveness of multimedia pre-training depends on careful implementation and sufficient infrastructure support (Abu Bakar et al., 2025; Lai & Chen, 2025).

The following is a brief table of Themes and Codings to illustrate the main findings:
 Theme Coding

Table 1: Themes and Codings to illustrate the main findings

Theme	Coding
Teaching Challenges	Difficulty understanding basic concepts, low engagement, limited retention, differences in student levels, teaching pressure
Existing Teaching Strategies	Step-by-step demonstrations, practical exercises, visual aids, collaborative learning, effectiveness depends on student level and resources
Perceptions of Computational Thinking (CT)	Helps to deconstruct problems, recognize patterns, build solution logic, challenges of independent application, need for step-by-step guidance
Role of Multimedia Pre-Training	Helps with initial understanding, increases engagement, supports retention, logistical constraints, student discipline, resource dependency

From the analysis, main observations show that each theme is interconnected with each other. Teaching challenges encourage lecturers to use existing teaching strategies to address students' difficulties. Positive perceptions of CT provide motivation to integrate computational thinking into teaching strategies, while multimedia pre-training is seen as a supporting tool that can improve overall teaching effectiveness. Coding analysis also shows that the effectiveness of this strategy is highly dependent on lecturer experience, classroom context, and available resources.

In addition, the themes that emerged also show that lecturers are aware of the potential of pedagogical innovation, but their implementation faces practical challenges. The Teaching Challenge and Multimedia Pre-Training themes emphasize that even when new strategies are introduced, the real context of the classroom and student discipline determine the extent to which the strategy is successful. The CT Perception theme shows that lecturers actively consider the value of pedagogy to improve students' problem-solving skills. Existing Teaching Strategies also emphasize that although innovation is needed, traditional approaches are still widely used to support student learning.

In conclusion, the main results and observations show a consistent pattern: the themes that emerged from the initial coding provide a comprehensive picture of the reality of programming pedagogy in matriculation colleges, the challenges faced, the strategies used, and the lecturers' perceptions of innovative approaches such as CT and multimedia pre-training. These findings form the basis for further discussion on the effectiveness of integrating innovative pedagogical strategies and the factors that influence their success in real classroom contexts.

This study provides a comprehensive overview of lecturers' perspectives on teaching programming in matriculation colleges, particularly in relation to the integration of Computational Thinking (CT) and Multimedia Pre-Training. Supporting evidence was obtained from interviews and observations conducted with five participants. Lecturers consistently stated that one of the main challenges was that students had difficulty understanding basic programming concepts, which in turn affected their engagement. For example, Participant 1 stated, "Mmm... aaa, sometimes students have difficulty understanding basic concepts eh... some get bored quickly," while Participant 3 added, "Mmm... haa, only a little engagement... eh... some lose focus when coding is long." This dialogue reinforced the theme of Teaching and coding Challenges related to understanding basic concepts and low student engagement.

In addition, the teaching strategies used by lecturers, such as step-by-step demonstrations and practical exercises, were also reinforced through Participant 2's dialogue, which stated, "Aaa... mmm, for a lot of practical exercises, eh, let students try coding themselves." This Existing Teaching Strategy Theme shows that a more hands-on approach helps overcome some of the challenges students face, although its effectiveness varies according to individual ability levels.

In the discussion of key findings, the study findings highlighted that lecturers see Computational Thinking as an effective pedagogical tool to build students' problem-solving skills. Participant 1 stated, "Mmm... aaa, good eh... can teach students to think systematically," while Participant 3 added, "Mmm... aaa, he can help students see patterns... eh, but needs guidance." This dialogue shows that CT not only improves students' ability to decipher problems and recognize patterns, but also emphasizes the importance of step-by-step guidance so that students can apply them effectively. This theme emphasizes that the integration of CT in programming teaching is not only positively received, but also requires careful implementation strategies.

The role of Multimedia Pre-Training is also reinforced through the dialogues of Participant 3, "Mmm... aaa, good eh, dynamic visuals make students more engaged," and Participant 5, "Eh... mmm, okay but aaa, students need to be disciplined to use the app." This theme shows that pre-training materials can increase students' engagement and initial understanding, but their effectiveness depends on student discipline and the availability of resources. It provides additional support to the theme of Existing Teaching Strategies, where multimedia can function as a learning reinforcer before the actual class.

Unexpected or contradictory findings emerged when several lecturers stated that although multimedia pre-training was beneficial, logistical constraints such as lack of computers or internet connection were practical obstacles. Participant 2 stated, "Eh... mmm, ok je... aaa, tapi kadang technical problem tu jadi hindi." This shows that although technology is considered helpful, the reality of implementation in the classroom can pose unexpected challenges. In addition, although CT is viewed positively, some students still have difficulty applying this principle independently, indicating that the feasibility of the pedagogical strategy depends on the support and context of the actual teaching.

In conclusion, the findings of the study emphasize that teaching programming in matriculation colleges faces challenges in terms of understanding basic concepts and student engagement. Lecturers use existing strategies and see the potential for integrating CT and multimedia pre-training as an effective approach. However, factors such as student discipline, resource availability, and logistical challenges affect the effectiveness of implementation. This analysis provides an important basis for understanding how themes and coding are formed from lecturers' experiences and how they can be used as a guide for more effective and contextual teaching strategies in the future.

The conclusion of this study highlights the perspectives of lecturers in teaching programming in matriculation colleges, particularly regarding the integration of Computational Thinking (CT) and Multimedia Pre-Training. Analysis of interviews and observations with five lecturers revealed four main themes that emerged from the participants' coding: Teaching Challenges, Existing Teaching Strategies, Perceptions of Computational Thinking, and the Role of Multimedia Pre-Training. These themes provide a comprehensive picture of the lecturers' experiences, challenges faced, and pedagogical strategies applied to support student learning. The main findings indicate that lecturers face difficulties in ensuring students' understanding

of basic programming concepts and maintaining their engagement, but they adapt existing teaching strategies such as step-by-step demonstrations, practical exercises, and the use of visual aids to address these challenges.

In the discussion of findings, the Teaching Challenges theme is closely related to the first objective of the study, which is to identify the main challenges of lecturers in teaching programming. Coding that emphasizes the difficulty in understanding basic concepts, low student engagement, and limited retention indicates that lecturers need to take a layered and flexible approach to meet the needs of students. The Existing Teaching Strategies theme, on the other hand, answers the second objective of the study, which is to explore the teaching approaches and strategies used. Coding shows that demonstrations, practical exercises, and the use of visual aids are the main mechanisms to help students understand programming problem solving, although their effectiveness varies according to the student level and classroom context. The Perception of Computational Thinking (CT) theme emphasizes that lecturers see the pedagogical value of CT in improving students' problem-solving skills, especially through problem decomposition, pattern recognition, and the development of algorithmic logic. This theme provides direct support to the third objective of the study, which focuses on the effectiveness and feasibility of CT-based applications in teaching. Participant coding shows that although CT is received positively, its effectiveness depends on gradual guidance and lecturer support.

The final theme, The Role of Multimedia Pre-Training, suggests that pre-training materials or applications can support students in building initial understanding and increasing engagement before the actual class. While coding shows great potential, challenges such as logistical constraints, student discipline, and resource dependency were also identified. This theme emphasizes that while technological innovations can enhance teaching effectiveness, successful implementation requires careful planning and institutional support.

The contribution of this study to the field of programming education is significant. First, this study provides a deeper understanding of lecturers' experiences in overcoming teaching challenges and the pedagogical strategies used. Second, this study contributes to the literature on the integration of Computational Thinking and multimedia pre-training in the context of matriculation, by emphasizing the role of lecturers as a key factor in ensuring the effectiveness of the strategy. Finally, the findings of this study provide practical guidance for the development of more integrated teaching applications, including recommendations for student engagement, multimedia use, and systematic implementation of CT.

In conclusion, this study confirms that the integration of CT and multimedia pre-training has the potential to improve student understanding and engagement in teaching programming, but its effectiveness is highly dependent on the lecturer's experience, teaching strategies used, and classroom context. The themes that emerged from the participants' coding provide a solid foundation for understanding the pedagogical challenges, effectiveness of strategies, and factors that support the success of programming learning at the matriculation level. This study not only provides academic insights, but also provides practical guidance for improving the quality of programming teaching in higher education institutions.

Based on your discussion and research findings, the appropriate model to address the problem statement, namely the challenge of teaching programming and the effectiveness of the Computational Thinking (CT) and Multimedia Pre-Training integration strategies, is the Integrated Pedagogical Strategy Integration model for Teaching Programming. This model is

specifically designed to combine lecturer experience, existing teaching strategies, CT, and multimedia pre-training to improve students' understanding of basic concepts, engagement, and problem-solving skills in matriculation colleges.

The formation of this model is based on themes and coding that have been analyzed from participant interviews. The four main themes that form the basis of the model are:

1. Teaching Challenges – Based on coding that emphasizes students' difficulty understanding basic concepts and low engagement, this model emphasizes identifying early challenges and providing appropriate support strategies.
2. Existing Teaching Strategies – The model combines demonstration approaches, practical exercises, and visual aids, which have been identified as effective mechanisms by lecturers to help students understand programming problem solving. This strategy is a basic element in the model to ensure effective teaching implementation.
3. Computational Thinking (CT) Integration – This theme is included as an innovative pedagogical element in the model. CT allows students to systematically solve problems, recognize patterns, and build algorithmic logic. The model emphasizes step-by-step guidance and the provision of activities that can train students to apply CT practically.
4. Pre-Training Role of Multimedia – The model supports the use of interactive materials or applications before the actual class to increase initial understanding and student engagement. This theme helps provide a foundation before students face more complex conceptual challenges.

This model is formed in a layered and interactive manner, where each element is interconnected. The teaching challenge serves as a starting point to identify student needs. Existing teaching strategies provide practical daily teaching mechanisms. CT and multimedia pre-training serve as pedagogical support elements to increase student engagement and build problem-solving skills. In its implementation, this model emphasizes active interaction between lecturers and students, where the lecturer acts as a facilitator who guides students through practical activities, problem decomposition, and the use of multimedia applications. In addition, this model takes into account classroom context factors and existing resources, such as access to computers, internet connections, and student discipline in using applications. By emphasizing this integrated integration, the model can solve the study's problem statement, which is to overcome the challenges of teaching programming, increase engagement, and strengthen students' understanding of basic concepts through innovative and contextual pedagogical approaches.

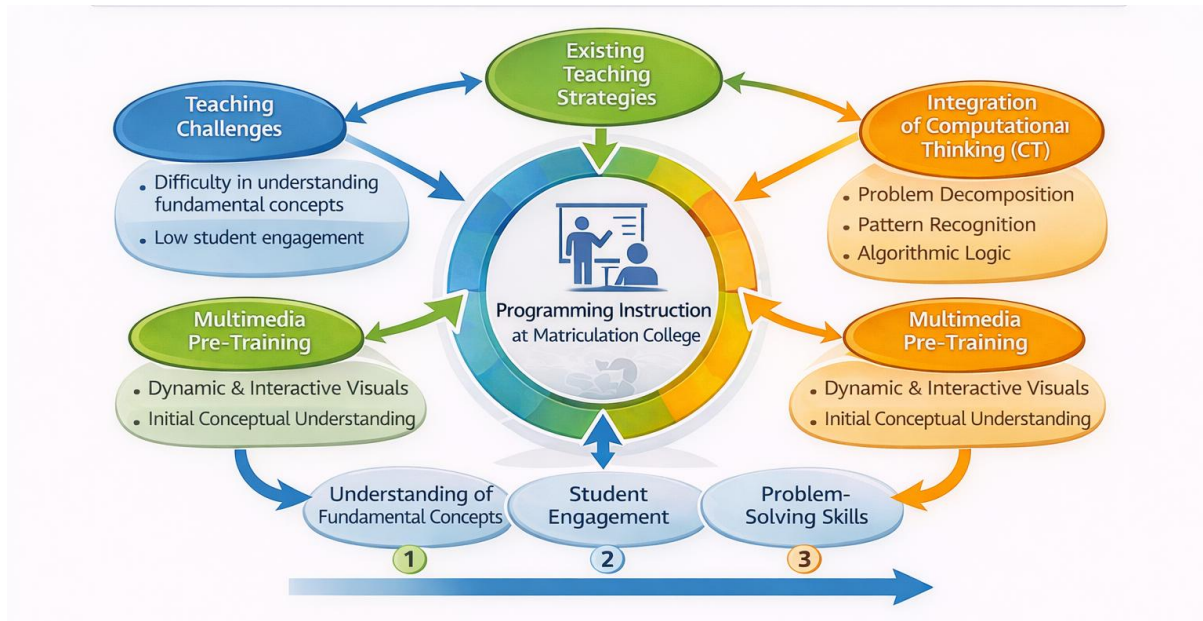


Figure 3: Model Integrated Pedagogical Strategy for Programming Instruction

Conclusions

This study explores lecturers' perspectives on the challenges of teaching programming and the potential for integrating Computational Thinking (CT) and pre-training multimedia in matriculation colleges. Findings indicate that students still face difficulties in understanding basic concepts, building algorithmic reasoning, and maintaining engagement, even though the syllabus has been fully delivered. These challenges are influenced by the diversity of students' backgrounds and teaching time constraints.

Thematic analysis identifies four main themes: Teaching Challenges, Existing Teaching Strategies, Perceptions of CT, and the Role of Pre-Training Multimedia. Conventional strategies such as demonstrations and practical exercises are still dominant but are not sufficient to support all students. Lecturers view CT as an effective pedagogical framework to support systematic problem solving, but require step-by-step guidance. Pre-training multimedia has the potential to increase students' initial understanding and engagement, in line with the principles of multimedia learning, but its effectiveness depends on the implementation context and institutional support. This study contributes to the literature by emphasizing the role of lecturers as a critical factor in the implementation of programming pedagogical innovations at the pre-university level.

Limitations

This study is limited to the perspectives of five lecturers at several matriculation colleges, making the scope of the data limited and not representative of the entire population. In addition, qualitative methods rely on interviews and observations, which may be influenced by lecturer subjectivity and researcher interpretation.

Research Implications

The findings of this study provide opportunities for future researchers to explore the effectiveness of integrating Computational Thinking and multimedia pre-training in the context of programming teaching in a broader, quantitative and longitudinal manner. This study also emphasizes the importance of assessing contextual factors such as classroom resources, student level, and application usage discipline. These findings can serve as a basis for developing a

more robust integrated pedagogical model that can be tested in various higher education institutions, thus adding to the literature on innovative strategies to enhance students' understanding, engagement, and problem-solving skills.

Practical Implications

Lecturers can use the findings of this study to design integrated teaching strategies with CT and multimedia pre-training. It allows them to tailor the approach to students' level, increase engagement, and reinforce understanding of basic concepts, while effectively utilizing technology resources in matriculation classrooms.

Future Research Directions

Based on the findings and limitations of this study, future research is recommended to involve larger and more diverse sample sizes, including lecturers from various institutions and teaching backgrounds, to increase the generalizability of the findings. Quantitative or mixed-methods studies can be conducted to assess the actual impact of integrating Computational Thinking and multimedia pre-training on students' academic achievement, engagement, and problem-solving skills. In addition, longitudinal studies can track changes in students' performance and attitudes over time. Researchers can also explore the use of new technologies or more sophisticated interactive applications to support programming instruction, as well as evaluate the effectiveness of integrated pedagogical models developed based on the findings of this study.

Acknowledgement

The author would like to express his sincere gratitude to all those who have contributed, directly and indirectly, to the completion of this study. Special appreciation goes to the lecturers who were willing to participate in the interviews and share their valuable experiences and insights. The author also expresses his gratitude for the support and cooperation given by the matriculation colleges involved in this research. Finally, sincere gratitude goes to all the individuals who have provided guidance, encouragement, and assistance to the successful completion of this study.

Conflict of Interest Statement

The authors declare that there are no conflicts of interest related to the publication of this study.

References

- Abu Bakar, E. E., Abd Halim, N. D., Abdul Hanid, M. F., & Inderawati, R. (2025). The challenges of teaching and learning programming in schools: Insights from a systematic literature review. *Karya Journal of Emerging Technologies in Human Services*, 1(1), 48–63. <https://karyailham.com.my/index.php/kjeths/article/view/203>
- Balacuit, I. C., Buan, A., Medina, J., & Oledan, A. M. (2024). Incorporating computational thinking in mathematics through block-based programming: Effects on students' problem-solving skills. *International Journal of Science Education and Teaching*, 3(3), 97–112. <https://doi.org/10.14456/ijset.2024.9>
- Chang, C. K., Liang, J. C., & Tsai, C. C. (2024). The transfer effect of computational thinking (CT)-STEM: A systematic literature review and meta-analysis. *International Journal of STEM Education*, 11, 44. <https://doi.org/10.1186/s40594-024-00498-z>
- Chen, CM., Huang, MY. Enhancing programming learning performance through a Jigsaw collaborative learning method in a metaverse virtual space. *IJ STEM Ed*, 11, 36 (2024). <https://doi.org/10.1186/s40594-024-00495-2>

- Fagerlund, O., & Rissanen, P. (2024). Pedagogical challenges in programming education: Complexity, time constraints, and resource limitations. *International Journal of Computer Science Education in Schools*, 7(1), 1–XX. <https://files.eric.ed.gov/fulltext/EJ1454643.pdf>
- Gao, Z., Yan, H., Liu, J., et al. (2025). Tracing distinct learning trajectories in introductory programming course: A sequence analysis of score, engagement, and code metrics for novice computer science vs. math cohorts. *International Journal of STEM Education*, 12, 27. <https://doi.org/10.1186/s40594-025-00546-2>
- Lai, C.-Y., & Chen, L.-J. (2025). Effects of web-based multimedia annotation on the performance, self-regulation, and cognitive load of students. *Educational Technology & Society*, 28(2), 238–258. <https://www.jstor.org/stable/48827985>
- Liu, Z., Gearty, Z., Richard, E., Orrill, C. H., Kayumova, S., & Balasubramanian, R. (2024). Bringing computational thinking into classrooms: A systematic review on supporting teachers in integrating computational thinking into K-12 classrooms. *International Journal of STEM Education*, 11, 51. <https://link.springer.com/article/10.1186/s40594-024-00510-6>
- Chen, CM., Huang, MY. Enhancing programming learning performance through a Jigsaw collaborative learning method in a metaverse virtual space. *IJ STEM Ed*, 11, 36 (2024). <https://doi.org/10.1186/s40594-024-00495-2>
- Mayer, R. E. (2022). Cognitive theory of multimedia learning. In R. E. Mayer (Ed.), *The Cambridge handbook of multimedia learning* (3rd ed.). <https://www.researchgate.net/publication/358123456>
- Mayer, R. E. (2024). The past, present, and future of the cognitive theory of multimedia learning. *Educational Psychology Review*, 36(1), 8. <https://doi.org/10.1007/s10648-023-09842-1>
- Omeh, C. B., Ayanwale, M. A., Mnguni, L. E., et al. (2025). Fostering programming skill and critical thinking through AI-assisted PBL integration. *Journal of New Approaches in Educational Research*, 14, 22. <https://doi.org/10.1007/s44322-025-00041-0>
- Perez Valdes, G. P., et al. (2025). Integrating computational thinking in children aged 3 to 6: Challenges and future directions. *Frontiers in Education*. <https://www.frontiersin.org/articles/10.3389/educ.2025.1535135>
- Sala Hudin, S. (2023). A systematic review of the challenges in teaching programming for primary school students. *Online Journal for TVET Practitioners*, 8(1), 75–88. <https://publisher.uthm.edu.my/ojs/index.php/oj-tp/article/view/13350>
- Selamat, S. M., Nasir, M. K. M., & Adnan, N. H. (2024). Investigation of computational thinking skills through instructional techniques, games and programming tools. *International Journal of Learning, Teaching and Educational Research*, 23(10), 435–452. https://ijlter.org/index.php/ijlter/article/download/11390/pdf?utm_source=web_search
- Sibia, N., et al. (2025). From code to concept: Evaluating multiple coordinated views in introductory programming. *arXiv*. <https://arxiv.org/abs/2501.12345>
- Ubaidullah, N. H., et al. (2023). Improving novice students' computational thinking skills by problem-solving and metacognitive techniques. *International Journal of Learning, Teaching and Educational Research*. <https://ijlter.net>
- Wing, J.M. (2006) Computational Thinking. *Communications of the ACM*, 49, 33-35. <https://doi.org/10.1145/1118178.1118215>