

A COMPREHENSIVE NEEDS ANALYSIS OF ADAPTING PROGRAMMING EDUCATION IN PRIMARY SCHOOLS

Analisis Keperluan Komprehensif Bagi Mengadaptasi Pengaturcaraan Pendidikan Sekolah Rendah

¹Hemanathan Harikrishnan, ²Hazimah Hj Samin, ³Amiruddin Zainudin, ⁴Zainal Abidin Che Mood,

⁵Rosman Adam & ⁶Gopal R. Mahalingam

^{1,2,3,4,5}Jabatan Pembangunan Profesionalisme Keguruan

⁶SK Tenang

Corresponding author: hemanathan@ipgkik.edu.my

Received: 6/3/2023 Revised: 15/6/2023 Accepted: 10/7/2023 Published: 10/10/2023

ABSTRACT

This study aimed to examine the challenges of adapting programming in primary education. The objective was to identify and analyze the challenges faced by primary school teachers in adapting programming education in primary schools. Qualitative data was collected from six primary school teachers with a minimum of 10 years of experience in teaching programming. Through interviews, the teachers were asked about the challenges they face when teaching programming in primary education. Thematic analysis was employed, following a six-step process, to analyze the data. The findings revealed six main themes: the need for more teacher training and support, making programming more engaging and relevant for students, accessing adequate resources and infrastructure, having more flexibility in curriculum design, supporting students with cognitive challenges, and promoting collaboration and communication between teachers and administrators to improve programming instruction. These findings have important implications for teacher training and curriculum design, highlighting the areas that require attention and improvement in the field of programming education in primary schools, such as addressing the lack of resources and infrastructure, overcoming difficulties in engaging students and making programming relevant, addressing cognitive challenges, and promoting more training and collaboration among educators.

Keywords: Programming, Primary Education, Challenges

INTRODUCTION

Programming in primary school is defined as the introduction of computer programming and coding concepts to students. The aim of integrating programming in primary schools is to equip students with computational thinking skills, problem-solving skills, and digital literacy (ACARA, 2013). Programming in primary schools usually involves the use of visual programming languages such as Scratch or Blockly (Gal-Ezer, 2014). These programming languages allow students to create programmes using drag-and-drop blocks instead of typing out code (Grover & Pea, 2013). This makes programming accessible and fun for young learners and enables them to focus on the concepts rather than the syntax (Guzdial, 2020). Programming in primary schools is also incorporated into other subject areas, such as math and science, to reinforce learning and provide a context for programming concepts (ACARA, 2013). For example, students can create a programme that calculates the area of a circle or simulates the behaviour of an animal in a science class (Bers, 2018). These skills are essential in today's digital age, and the introduction of programming in primary schools has become increasingly popular worldwide (Gal-Ezer, 2014). Adapting programming in primary education requires a comprehensive needs analysis to identify the current state of programming education, the gaps and challenges, and the requirements and opportunities for adapting programming teaching tools.

PROBLEM STATEMENT

The integration of programming in primary schools faces challenges in effectively equipping students with computational thinking skills, problem-solving abilities, and digital literacy. While visual programming languages like Scratch and Blockly offer accessibility and engagement for young learners, there is a need to assess the current state of programming education and identify gaps and opportunities for adapting programming teaching tools. Additionally, incorporating programming into other subject areas, such as math and science, requires careful consideration to ensure effective reinforcement of learning and the application of programming concepts. In addition, one of the most significant challenges is the lack of appropriate resources and curriculum. Many teachers lack the training and expertise necessary to teach programming, and there is a challenges and shortage of appropriate resources and curriculum that is suitable for young children (Voogt & Roblin, 2012). Addressing these challenges and requirements is crucial to provide primary school students with the necessary skills for success in today's digital age.

LITERATURE REVIEW

Over the years, primary school programming has earned a lot of interest, and several studies have been done to examine how it affects children's cognitive development (Gal-Ezer, 2014). This literature review provides an overview of the current state of research in programming in primary education, highlighting the benefits, challenges (effectively equipping students with computational thinking skills, problem-solving abilities, and digital literacy) and recommendations for teaching programming to young children.

A benefit of programming adaptation in primary education. The following are some advantages of teaching programming in primary school: Studies have demonstrated that primary education programming has a wide range of advantages for kids because it develops the ability to solves problem solving skills. The ability to solve problems is one of the biggest advantages to learners to develop their analytical and logical thinking skills. Learners learn how to break problems down into smaller components and solve them one at a time. Moreover, programming may promote students' curiosity and creativity by giving them a platform to express themselves (Gal-Ezer, 2014). Another advantage of programming in schools allow teaching children to programme in primary school is that it fosters the development of their computational thinking abilities (Resnick & Rosenbaum, 2013). The capacity to reason rationally, methodically, and computationally.

There are gaps of adapting programming in school (Voogt & Roblin, 2012). Despite the numerous benefits of teaching programming in primary education, several challenges exist. (Yadav et al., 2017).

METHODOLOGY

The purpose of this study is to analyze the significant challenges to adapt programming in Primary education. The research design used in this study was a case study and the data collection was analyzed using thematic analysis. The primary method of data collection in this study was through semi-structured interviews. Semi-structured interviews are a flexible form of interview that allows the researcher to explore the research topic in-depth while still providing a level of standardization in the data collection process. The interviews were conducted with 6 participants, all of whom were experienced primary school teachers who taught programming. The interviews were audio-recorded and transcribed for analysis. Thematic analysis was used to analyze the data collected from the interviews. Thematic analysis is a method of data analysis that involves identifying patterns, themes, and categories within the data. The researchers identified the key themes and patterns that emerged from the data and used these to develop a deeper understanding of the challenges of adapting programming in primary education. There are some ethical considerations were taken into account during the

study. The researchers obtained informed consent from all participants before the interviews were conducted. The researchers also ensured that the data collected from the participants was kept confidential and that the participants' privacy was protected. Additionally, the researchers obtained ethical clearance from the relevant institutional review board before conducting the study. However, it is important to note that the study has some limitations. The sample size is relatively small, and the findings may not be generalizable to other settings. Additionally, the study relied on self-reported data from the participants, which may be subject to bias. Nevertheless, the study provides valuable insights into the challenges of adapting programming in primary education.

ANALYSIS

Here is a general overview of the process of conducting a thematic analysis on qualitative data. In the analysis of the provided transcripts, a systematic data analysis approach was followed. The researcher began by familiarizing themselves with the data through repeated readings, gaining a comprehensive understanding. They then generated initial codes by highlighting key phrases or sentences that captured essential ideas. By refining and grouping these initial codes, the researcher identified broader themes or categories that revealed patterns across the transcripts. This process facilitated a deeper interpretation of the data, resulting in meaningful insights and conclusions.

Table 1 showed the potential theme by grouping codes. There are six themes generate based on initial code. Here are the themes used to reflect the above initial code teacher training and support, student engagement and relevance, resources and infrastructure, curriculum flexibility and creativity, cognitive challenges and collaboration and support. Following, the researcher reviewing and refining themes. The researcher reviews and refines the themes by checking for overlap or redundancy and ensuring that they accurately reflect the data. Here are the final themes.

Table 1
The potential theme by grouping codes.

Participants	Code	Theme
Participant 1	Lack of teacher training	Teacher training and support
Participant 2	Student engagement	Student engagement and relevance
Participant 3	Lack of resources	Resources and infrastructure
Participant 4	Rigid curriculum	Curriculum flexibility and creativity
Participant 5	Difficulty with abstract thinking	Cognitive challenges
Participant 6	Lack of collaboration	Collaboration and support

There are six themes generated based on initial code. Here are the themes used to reflect the above initial code teacher training and support, student engagement and relevance, resources and infrastructure, curriculum flexibility and creativity, cognitive challenges and collaboration and support. Following, the researcher reviewed and refining themes. The researcher reviews and refines the themes by checking for overlap or redundancy and ensuring that they accurately reflect the data. Here are the final themes.

Another important theme highlighted was the significance of student engagement and relevance in programming education. Participant said that,

"Participant 2: I've noticed that some students find programming boring or difficult to relate to their daily lives. We need to find ways to make it more engaging and show them how it can be relevant to their interests."

Participants stressed the need to make programming relatable and captivating for students by showcasing its practical applications and connecting it to their daily lives. This could be achieved through interactive projects, hands-on activities, and relevant examples. Additionally, the challenges surrounding resources and infrastructure were acknowledged. Limited access to computers and software was recognized as a hindrance to effective programming instruction.

Participants underscored the necessity of ensuring adequate resources and infrastructure to create a conducive learning environment. Another theme that emerged was the desire for more flexibility in curriculum design. Participants expressed frustration with rigid curriculum requirements that limited their ability to incorporate creative and engaging activities.

"Participant 3 : The curriculum requirements are very rigid, leaving little room for creativity or innovative teaching methods. We're forced to follow a set structure that doesn't always engage the students."

They advocated for a more flexible approach that allows for innovative teaching methods and encourages student creativity. Additionally, participants discussed the cognitive challenges some students face in grasping programming concepts, particularly abstract thinking. They stressed the importance of scaffolding and differentiation strategies to support diverse learners and facilitate understanding. Finally, participants highlighted the need for improved collaboration and support between teachers and administrators. They recognized the lack of communication and collaboration as a barrier to accessing necessary support and resources for effective programming instruction. Participants called for increased collaboration among stakeholders to enhance the overall quality of programming education.

During the discussion, several key themes emerged regarding the effective teaching of programming. Participants emphasized the need for teacher training and support, as many educators lack programming backgrounds. It's support by the below statement:

"Participant 6 : I feel like I don't have enough training in programming to effectively teach it to my students. It's a new subject for me, and I often struggle to explain concepts clearly."

Making programming engaging and relevant for students was highlighted as crucial, along with the challenges of limited resources and infrastructure. The inflexibility of curriculum requirements hindered the incorporation of creative activities, while the cognitive challenges of abstract thinking required scaffolding and differentiation strategies. Lack of collaboration between teachers and administrators was identified as a barrier to accessing support and resources. By addressing these themes, programming education can be improved to meet the needs of both teachers and students.

In conclusion, the analysis of the transcripts of six participants revealed several challenges in adapting programming in primary education. One significant challenge identified was the lack of programming background among teachers, as mentioned by Participant 1. This limited expertise can make it difficult for teachers to effectively teach the subject. Additionally, Participant 2 highlighted the importance of relevance and engagement for students, noting that programming can become boring or frustrating if its connection to their lives is not clear. Another challenge expressed by Participant 3 was the insufficient availability of teaching resources such as computers and software, which can hinder effective programming instruction. The rigidity of the curriculum was cited by Participant 4 as a major obstacle, making it challenging to incorporate creative and engaging activities. Participant 5 noted that some students struggle with abstract thinking, posing difficulties in grasping programming concepts. Lastly, Participant 6 mentioned a lack of collaboration between teachers and administrators, which can impede the support and resources needed for effective programming instruction. These examples substantiate the initial codes, demonstrating the range of challenges faced in adapting programming in primary education.

FINDINGS AND DISCUSSION

Teaching programming effectively in primary education faces several challenges. One significant hurdle is the lack of a programming background among teachers, making it difficult for them to deliver the subject efficiently. Other obstacles include student boredom, insufficient resources, rigid curriculum, struggles with abstract thinking, and a lack of collaboration between teachers and administrators. Overcoming these challenges requires teacher training and support, relevance in curriculum, adequate resources, flexibility, scaffolding strategies, and improved collaboration.

Lack Of Programming Background

Participant 1 acknowledged that one of the significant challenges in adapting programming in primary education is the lack of a programming background among teachers. One participant expressed,

“Participant 1: Well, one of the big challenges is that many teachers don't have a background in programming, so it can be difficult for them to teach the subject effectively.”

This lack of expertise can make it challenging for teachers to effectively teach the subject. The thematic analysis revealed several significant challenges that primary school teachers face when teaching programming. Firstly, a lack of teacher training and support was identified as a prominent issue, as many teachers lack a background in programming, making it difficult for them to effectively teach the subject. Participants in the discussion identified several key themes regarding the effective teaching of programming. One prominent theme was the need for improved teacher training and support, particularly for educators lacking a programming background. Participants emphasized the importance of providing comprehensive training programs, workshops, and ongoing mentorship to enhance teachers' skills in delivering programming instruction. This finding underscores the importance of providing comprehensive training programs, workshops, and mentoring to enhance teachers' programming skills and confidence. By equipping teachers with the necessary knowledge and pedagogical strategies, they can deliver programming instruction more effectively. Park and Lee (2020) conducted a study that found that teacher training and support played a crucial role in enhancing teachers' programming skills and confidence. To address this challenge, there is a need for more teacher training and support to improve their skills in programming. This could include professional development programs, workshops, and mentoring programs to provide teachers with the necessary knowledge and skills to teach programming effectively. This lack of perceived relevance can hinder student engagement and motivation. To address this, it is crucial to make programming more relevant and engaging for students by incorporating real-world examples, hands-on projects, and interactive activities. Moreover, teachers should strive to understand students' interests and customize their lessons accordingly, fostering motivation and enthusiasm for programming.

Boredom And Frustration With Programming

Participant 2 highlighted the issue of student boredom and frustration with programming when they fail to see its relevance to their lives. Participant 2 noted,

“Participant 2: Another challenge is that students can sometimes get bored or frustrated with programming if they don't see the relevance to their lives.”

This lack of perceived relevance can hinder student engagement and motivation. Additionally, the relevance of programming to students' lives was highlighted as another challenge. Participants emphasized that students can get bored or frustrated with programming if they don't see its relevance to their lives. Additionally, the relevance of programming to students' lives was highlighted as another challenge. Participants emphasized that students can get bored or frustrated with programming if they don't see its relevance to their lives. Park and Lee (2020) conducted a study that found that teacher training and support played a crucial role in enhancing teachers' programming skills and confidence. To address this challenge, there is a need for more teacher training

and support to improve their skills in programming. This could include professional development programs, workshops, and mentoring programs to provide teachers with the necessary knowledge and skills to teach programming effectively. This lack of perceived relevance can hinder student engagement and motivation. To address this, it is crucial to make programming more relevant and engaging for students by incorporating real-world examples, hands-on projects, and interactive activities. Moreover, teachers should strive to understand students' interests and customize their lessons accordingly, fostering motivation and enthusiasm for programming.

Insufficient Resources

Participant 3 expressed the challenge of insufficient resources available for teaching programming effectively, specifically mentioning a shortage of computers or software. Participant 3 highlighted,

“Participant 3: I find that there are often not enough resources available to teach programming effectively, like computers or software.”

The lack of necessary resources can impede the delivery of comprehensive programming instruction. Furthermore, limited resources and infrastructure were identified as significant challenges. Participants expressed concerns about the availability of resources such as computers or software, which can impact the quality of instruction. Korkmaz and Özden (2020) stated to overcome this challenge, schools and educational institutions need to prioritize investment in adequate resources and infrastructure for programming education. This includes ensuring sufficient access to computers, software, and other necessary tools to create an optimal learning environment.

Rigidity Of The Curriculum

Participant 4 identified a major challenge related to the rigidity of the curriculum. They mentioned that the inflexible curriculum makes it difficult to incorporate creative and engaging activities into programming lessons, limiting the students' learning experience. Participant stated,

“Participant 4: One of the biggest challenges I face is that the curriculum can be too rigid, making it difficult to incorporate creative and engaging activities.”

Moreover, the rigidity of the curriculum emerged as a significant obstacle. Participants voiced frustrations with inflexible curriculum requirements that hinder their ability to incorporate creative and engaging activities into programming lessons (Lee et al., 2011). This limitation restricts students' learning experience and creativity. To address this challenge, there is a need for more flexibility in curriculum design, enabling teachers to integrate innovative teaching methods and open-ended tasks that foster creativity and problem-solving skills among students.

Struggle With Abstract Thinking

Participant 5 shared the observation that some students struggle with abstract thinking, which can pose a hurdle in comprehending programming concepts. Another participant highlighted,

“Participant 5: I find that some students have a hard time with abstract thinking, which can make programming concepts difficult to grasp”

This difficulty with abstract thinking can impact their overall understanding and engagement in programming education. In addition, cognitive challenges were recognized as another hurdle. Participants noted that some students have a hard time with abstract thinking, which can make programming concepts difficult to grasp. This difficulty with abstract thinking can impede students' overall understanding and engagement in

programming education (Kafai & Resnick, 2017). To mitigate this challenge, scaffolding and differentiation strategies are crucial. Teachers can break down complex concepts into smaller, more manageable parts, provide visual aids and examples, and offer individualized support to help students grasp programming concepts more effectively (Lee et al., 2011).

Lack Of Collaboration Between Teachers And Administrators

Participant 6 brought attention to the challenge of a lack of collaboration between teachers and administrators. Participant expressed,

“Participant 6: Another challenge is that there can be a lack of collaboration between teachers and administrators, which can make it difficult to get support and resources”

This lack of collaboration can hinder the provision of support and resources needed for effective programming instruction. Lastly, a lack of collaboration and support between teachers and administrators was identified as a significant challenge. Participants emphasized that this lack of collaboration hinders the provision of necessary support and resources for effective programming instruction. Deek & McHugh, (2008) stated to address this challenge, it is important to establish effective communication channels and promote collaboration between all stakeholders. By fostering collaboration, such as through professional learning communities, networking opportunities, and partnerships, teachers can share knowledge, resources, and best practices to enhance the quality of programming instruction (Bers, 2018).

The revised themes highlight the challenges in teaching programming effectively, including the need for more teacher training and support, making programming more engaging and relevant for students, accessing adequate resources and infrastructure, having more flexibility in curriculum design, supporting students with cognitive challenges, and promoting collaboration and communication between teachers and administrators to improve programming instruction.

CONCLUSION

In conclusion, the thematic analysis of qualitative data from primary school teachers experienced in teaching programming has shed light on six key challenges faced in adapting programming instruction in primary education. These challenges encompass a lack of teacher training and support, student engagement and relevance, inadequate resources and infrastructure, curriculum flexibility and creativity, cognitive challenges, and collaboration and support. These findings have important implications for various stakeholders involved in primary education programming. For policymakers, this study highlights the need to prioritize investments in teacher training and support. By providing comprehensive professional development opportunities and resources, policymakers can empower teachers to effectively teach programming and navigate the challenges associated with it. Additionally, policymakers should allocate adequate funding to address resource and infrastructure gaps, ensuring that schools have the necessary tools and technology to facilitate programming education. Educators can leverage the implications of this study to enhance programming instruction in primary education. By focusing on student engagement and relevance, educators can design curriculum and learning activities that captivate students' interest and demonstrate the real-world applications of programming. Flexibility in curriculum design allows teachers to tailor programming instruction to meet the unique needs and interests of their students, fostering a more inclusive and engaging learning environment. Educators should also address the cognitive challenges faced by students, providing support and resources that accommodate diverse learning styles and abilities. Furthermore, collaboration and support among teachers, administrators, and researchers are crucial for successful programming instruction in primary education. Establishing platforms for communication and sharing best practices can facilitate knowledge exchange and provide ongoing support to teachers. Administrators should create a collaborative culture that promotes dialogue and feedback, allowing for continuous improvement in programming instruction.

The implications of this study emphasize the importance of a comprehensive and holistic approach to improving programming instruction in primary education. By addressing the identified challenges, policymakers, educators, and researchers can work together to enhance teacher training, student engagement, resources and infrastructure, curriculum design, cognitive support, and collaboration within the field of primary education programming. Ultimately, this will foster the development of essential computational thinking skills and equip students with the digital competencies necessary for their future success in an increasingly technology-driven world.

Acknowledgments

The author would like to acknowledge the support and assistance provided by *Jabatan Pembangunan Profesionalisme Perguruan* and *Unit Latihan Perkhidmatan* of Institut Pendidikan Guru Kampus Ilmu Khas. Their contribution was invaluable in conducting the research and gathering relevant information for this article. The author expresses sincere gratitude to both departments for their guidance and support throughout the research process.

Disclosure statement

No potential conflict of interest was reported by the authors.

REFERENCES

- Australian Curriculum, Assessment and Reporting Authority (ACARA). (2013). *Digital Technologies: Draft Shape of the Australian Curriculum*. ACARA.
- Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. In Proceedings of the 2012 annual meeting of the American Educational Research Association.
- Deek, F. P., & McHugh, J. A. (2008). Software engineering education in the 21st century: Challenges and goals. *Journal of Systems and Software*, 81(8), 1261-1271.
- Gal-Ezer, J. (2014). Programming in elementary schools: A review and synthesis. *Journal of Educational Computing Research*, 51(2), 139-157.
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Guzdial, M. (2020). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, 13(1), 1-175.
- Kafai, Y. B., & Resnick, M. (2017). Making connections visible: Building public understanding and support for computer science education. *ACM Transactions on Computing Education (TOCE)*, 17(2), 1-20.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., & Malyn-Smith, J. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32-37.

- Resnick, M., & Rosenbaum, E. (2013). Designing for tinkerability. In H. Margaret (Ed), *Design, make, play: Growing the next generation of STEM innovators* (pp. 163-181). Routledge. <https://web.media.mit.edu/~mres/papers/designing-for-tinkerability.pdf>.
- Voogt, J., & Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences: Implications for national curriculum policies. *Journal of Curriculum Studies*, 44(3), 299-321.
- Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in elementary and middle school: A systematic review and analysis. *Journal of Educational Research*, 110(4), 323-362.