Jurnal Ilmiah Profesi Pendidikan
Volume 10, Nomor 4b, Edisi Khusus 2025
ISSN (Print): 2502-7069; ISSN (*Online*): 2620-8326

Analysis of Physics Education Students' Computational Readiness in Modern Digital-Based Computational Learning

Muhammad Taufik*, Wahyudi, Fathurrahman
Program Studi Pendidikan Fisika, Jurusan PMIPA FKIP Universitas Mataram, Indonesia
*Corresponding Author: taufik@unram.ac.id

Article History
Received : September 16th, 2025
Revised : October 23th, 2025
Accepted : November 25th, 2025

Abstract: This study aims to provide a more in-depth description of Physics Education students' computational readiness for modern computational learning utilizing digital technology. Twenty-eight students participated as respondents, and data were collected through five main indicators: digital literacy, computational mathematics skills, programming experience, device readiness, and the level of difficulty in understanding computational concepts. The results indicate that student readiness is still in the moderate category, especially in digital literacy and mathematics skills. The majority of students also lack adequate programming experience, requiring more intensive guidance when dealing with program syntax and logic. The devices used by students vary in performance, which also affects the smoothness of computational practice. Furthermore, the level of difficulty of computational concepts is quite high, especially in reading pseudocode and debugging. Overall, the findings of this study demonstrate the need for a more adaptive, gradual learning strategy that takes into account the diversity of student abilities.

Keywords: Computing Readiness, Physics Education, Digital Literacy, Basic Programming, Modern Computing

INTRODUCTION

The integration of computing into physics education, which initially appeared merely as an additional enrichment component, has progressively evolved into an essential element within the curriculum. This development is not surprising, considering the increasing reliance on technological innovations across scientific fields, where computational tools have become indispensable for analyzing complex physical systems, performing numerical simulations, and managing large datasets that would be impractical to handle manually. As a result, computational competence is now widely recognized as an integral part of physics instruction, requiring students to develop not only conceptual mastery but also the practical ability to use algorithms, software applications, and programming languages as part of their routine academic work.

The importance of these competencies has been articulated in several recent studies. (Odden & Caballero, 2023) argue that computational literacy in physics education should be organized around a structured and systematic framework. Their work highlights the material, cognitive, and social dimensions of computational literacy, suggesting that a comprehensive integration of these dimensions is essential for both effective learning and meaningful engagement in physics research (Odden & Caballero, 2023). This perspective reflects a broader shift in the field, where computational literacy is increasingly regarded as a core component rather than a peripheral skill.

The relevance of computational integration becomes even more apparent when observed in the context of the transition from STEM to STEAM. By adding creative and artistic dimensions to traditional scientific and technological domains, STEAM approaches encourage students to engage in broader and more innovative forms of problem-solving. Deák and Kumar (2024) demonstrate that STEAM-based learning contributes to the development of digital competencies aligned with sustainable innovation, supporting the argument that computational integration in physics education can facilitate interdisciplinary thinking that better prepares students for emerging global challenges (Deák & Kumar, 2024).

This need for adaptation is also evident in the shifting landscape of instructional practice. (Hermawan et al., 2022) show that blended learning environments, when systematically integrated with digital tools, can improve students' literacy and problem-solving abilities, particularly in contexts that require continuous interaction with technological resources (Hermawan et al., 2022). Collectively, these studies suggest that a balanced combination of theoretical understanding and computational engagement has become a fundamental requirement for preparing students to navigate contemporary scientific and technological realities.

Despite this growing emphasis, student readiness for computational learning remains uneven. Many students enter higher education with widely varying levels of familiarity with digital technologies; while some are able to work comfortably with different applications, others may still struggle with basic functionalities. This disparity underscores the need for more differentiated pedagogical approaches that acknowledge these differing starting points (Mahligawati et al., 2023; Khan et al., 2025). Such variation indicates that computational readiness cannot be assumed to be uniform across cohorts.

Digital literacy constitutes one of the key factors contributing to this variability. It encompasses the ability to operate digital devices, manage information effectively, and navigate various types of applications. Studies indicate that students with higher levels of digital literacy transition more smoothly into new computational environments, while those with limited experience require additional support to understand fundamental interface features (Travero, 2022; Ziatdinov & Valles, 2022). Consequently, embedding digital literacy development within the curriculum has become increasingly important to ensure more equitable learning conditions (Awuor & Okono, 2022).

Another factor that strongly contributes to computational readiness is students' mathematical preparation, particularly in areas relevant to programming and numerical algorithms. Understanding mathematical functions, manipulating algebraic forms, and applying basic calculus concepts provide the conceptual foundation needed for computational tasks in physics (Zhang et al., 2023). Students who lack confidence in mathematics often find it challenging to interpret coding structures that represent mathematical operations, thereby revealing the close interplay between mathematical readiness and programming proficiency (Bitzenbauer et al., 2024).

Previous exposure to programming further influences students' readiness for computational learning. Those who have had prior experience in programming generally demonstrate greater ease in understanding logical reasoning and algorithmic design (Quang, 2025; Kemouss & Khaldi, 2025). Conversely, students without such experience may require substantial practice and guidance to establish foundational programming skills (Suson et al., 2020). This suggests that earlier exposure to programming can play an important role in strengthening students' confidence and preparedness for computational coursework.

In addition to these individual characteristics, the adequacy of students' devices serves as a practical but often overlooked component of computational readiness. Devices with lower specifications can hinder the use of essential software, leading to unequal learning conditions across students (Mercado, 2021; Nkweke, 2020). This reinforces the need to consider technological infrastructure as part of instructional planning to ensure that all students have access to computational resources required for learning (Robeva & Laubenbacher, 2009; Aly & Efendi, 2024).

Taken together, digital literacy, mathematical competence, programming experience, and device adequacy form a set of interconnected dimensions that shape students' overall readiness for computational learning. In light of these considerations, the present study aims to examine these factors among students in Physics Education programs. The insights derived from this investigation are expected to support the development of more adaptive and responsive pedagogical strategies that reflect the diverse readiness profiles of students, thereby contributing to more effective learning in computationally oriented physics courses (Gawade et al., 2022; -, 2023).

METHODS

The methods section of this study was designed to provide a clear overview of the approach used to capture students' computing readiness. Given that the research focused on mapping students' initial conditions, a quantitative descriptive approach was deemed most appropriate. This approach allows researchers to observe trends, variations, and patterns emerging from the data without intervening in the learning process.

This study adopted a quantitative descriptive design, which essentially aims to describe phenomena as they exist. This approach does not seek causal relationships, but rather describes actual conditions based on data obtained from respondents. This design was chosen based on the need to objectively understand students' readiness, particularly across five key indicators: digital literacy, computational mathematics skills, programming experience, device readiness, and the difficulty level of computing concepts.

The research participants consisted of 28 Physics Education students currently taking computing-related courses. All students were in their mid-terms and were considered to have sufficient prior experience using digital devices in their classes. Participants were selected through a total sampling method, considering the relatively small number of students in each class and their relevance to the research focus. Five instruments were used in this study, each designed to measure a more specific aspect of computing readiness.

a. Digital Literacy Test

This test consists of multiple-choice questions and two practical assignments that assess students' ability to operate digital devices, search for information, use common applications, and understand file structures. The questions are designed to meet the basic needs of computing learning.

b. Computational Mathematics Ability Test

This instrument contains 15 questions covering algebra, functions, trigonometry, and common mathematical operations frequently used in numerical algorithms. The questions are structured with varying levels of difficulty in mind to more accurately assess the diversity of student abilities.

c. Programming Experience Questionnaire

This instrument is simple, consisting of dichotomous (yes/no) questions regarding students' experience learning a specific programming language. The questionnaire also includes several open-ended questions to determine students' familiarity with programming structures and terms.

d. Device Readiness Scale

Students' device readiness is assessed using a Likert scale of 1–4. The assessment includes laptop or computer performance in running computing software, storage capacity, processing speed, and the device's ability to open specific compilers or IDEs.

e. Computing Concept Difficulty Diagnostic Test

This instrument aims to assess students' understanding of basic computing concepts such as pseudocode, logical flow, branching, looping, and debugging. The test includes multiple-choice and three-point questions that require students to identify errors in simple code. Data was collected during a single class session lasting approximately 90 minutes. Each student was asked to complete the instrument in a predetermined order. Monitoring was conducted to ensure no collaboration could compromise the authenticity of the data. All student responses were then collected and coded for quantitative analysis. Data analysis was performed using descriptive statistics, including mean values, standard deviations, and minimum and maximum values. Data were also visualized in the form of histograms, bar graphs, and scatterplots to more clearly identify trends. Furthermore, the relationship between variables was examined exploratively to provide an initial overview of possible influencing factors.

FINDINGS AND DISCUSSION

This section presents research findings based on five indicators of student computational readiness: digital literacy, computational mathematics skills, programming experience, device readiness, and the difficulty level of computational concepts. Data are presented in tables and various visualizations to provide a more comprehensive picture. The findings are then analyzed by considering the context of computing learning for Physics Education students.

Findings

Overview

In general, the five indicators show that students are still at a moderate level of readiness. None of the indicators show a particularly high level of readiness. In fact, some indicators such as programming experience and the difficulty level of computing concepts show a trend indicating the need for additional guidance. The following are descriptive statistics for the five readiness indicators.

Table 1. Descriptive Statistics of Five Computing Readiness Indicators

Variable	Mean	SD	Min	Max
Digital Literacy	65.4	9.0	48	83
Computational Mathematics Skills	59.8	10.8	40	84
Programming Experience*	0.25	–	0	1
Device Readiness	2.9	0.6	2	4
Computational Concept Difficulty	74.1	11.2	55	98

*Note: 0 = no experience, 1 = has experience.

Students with low-level devices (score 2) had a higher level of difficulty. Laptop performance affected the smoothness of the compiler.

Student Digital Literacy

The average digital literacy score was 65.4. This isn't a bad score, but it doesn't reflect truly mature digital skills, especially for computing-based learning. Some students appeared quite confident operating basic digital device features, such as searching for information and using common applications. However, when asked to use specialized applications or perform functions requiring file manipulation, some students appeared hesitant or took longer. The fact that digital literacy scores remained in the moderate category suggests that computing learning needs to begin with strengthening digital skills, particularly orientation toward the use of computing software.

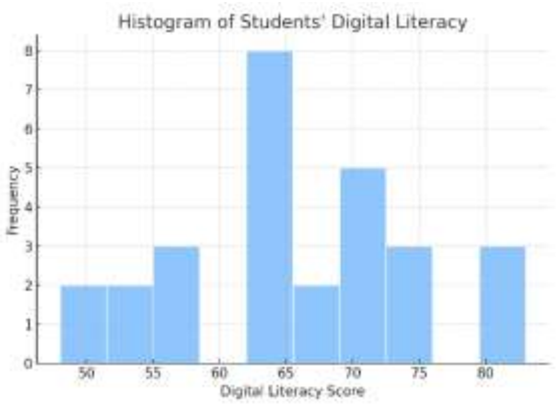

Figure 1. Histogram of Student Digital Literacy

Computational Mathematics Skills

Students' computational mathematics skills averaged 59.8, indicating that some students have not fully mastered the basic mathematical concepts required in computing. Upon closer analysis, students with low scores typically experienced difficulties with: understanding mathematical functions, algebraic manipulations, simple trigonometric calculations, reading equations. From a computational learning perspective, these results are crucial. The algorithmic logic underlying physics programming often requires the ability to decompose functions and interpret mathematical relationships. Without this foundation, students may struggle to understand the flow of code. These results require lecturers to reconsider the proportion of reinforcement of basic mathematics before students move on to numerical programming.

Student Programming Experience

Figure 2 shows a bar graph of student programming experience. The majority of students (75%) reported never having studied programming in any form. Most of them were only familiar with basic terms like "code," "loop," or "syntax," without any deeper understanding. Students who had studied programming (25%) had typically taken courses or courses on the basics of programming logic. This small group typically grasps pseudocode and simple program structures more quickly. The stark contrast between these two groups presents a challenge. When computing material is introduced, instructors often have to repeat basic concepts to ensure all students are on the same page.

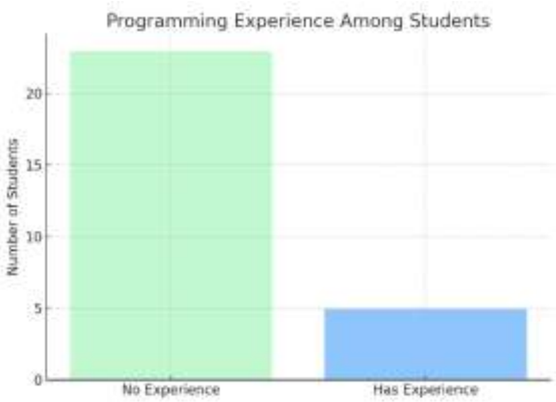

Figure 2. Bar Graph of Student Programming Experience

DOI: https://doi.org/10.29303/jipp.v10i4b.2452

As many as 75% of students have never studied programming. This explains the high rate of algorithmic errors appearing on the diagnostic test.

Student Device Readiness

Figure 3 illustrates the relationship between device readiness and computational difficulty. The average student device readiness score was 2.9 on a scale of 1–4. This indicates that most students have devices that are "sufficient" for basic computing learning. However, there is significant variation. Devices with minimal specifications often experience issues such as: slow execution times, difficulty opening applications, errors when installing the compiler, overheating when running simple simulations. This device readiness factor significantly impacts student motivation, especially those who are frustrated when their devices do not support their course requirements.

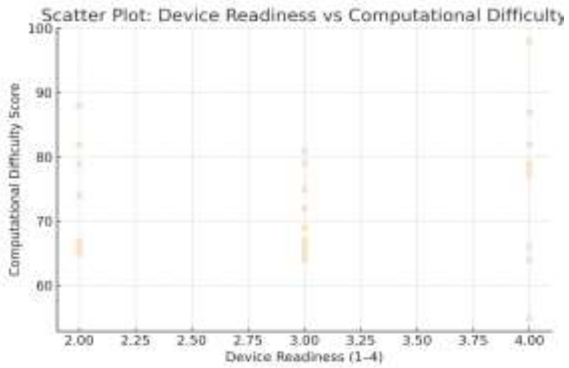

Figure 3. Scatter Plot of Device Readiness and Computational Difficulty

On the head and tail of the table. Font sizes for table and picture entries may be reduced.

Computing Concept Difficulty Level

The figure 4 displays the mean values for digital literacy, computational mathematics ability, device readiness (normalized to a 0–100 scale), and concept difficulty. Students demonstrated moderate levels of digital literacy and mathematical readiness, while device readiness showed considerable variation. Concept difficulty presented the highest mean score, indicating that students perceive computational concepts such as pseudocode, control structures, and debugging as significantly challenging.

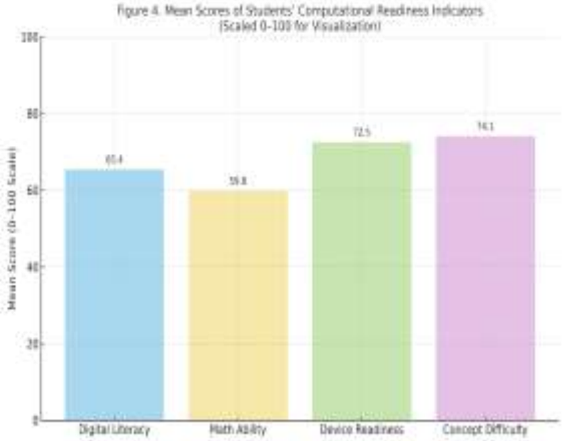

Figure 4. Mean scores of students' computational readiness indicators.

The average difficulty score for computing concepts is quite high (74.1). This means that many students have difficulty understanding: branching logic, loops, program flow, debugging. Some students reported being able to follow the explanations but felt confused when they had to implement them themselves. This indicates that students need more intensive practical experience. Figure 5 presents the mean concept difficulty scores for the two student groups based on prior programming experience. Students without programming experience show a substantially higher level of perceived difficulty compared to their peers who have previously engaged in programming tasks. The error bars indicate the standard deviation within each group, showing a wider variation among students with no programming experience. This pattern suggests that prior exposure to programming plays a meaningful role in reducing cognitive barriers when interacting with core computational concepts such as pseudocode, loops, conditionals, and debugging. The figure clearly demonstrates that students who have at least minimal programming experience tend to approach computational tasks with lower perceived difficulty and more stable performance, reinforcing the importance of foundational programming instruction in physics education contexts.

DOI: https://doi.org/10.29303/jipp.v10i4b.2452

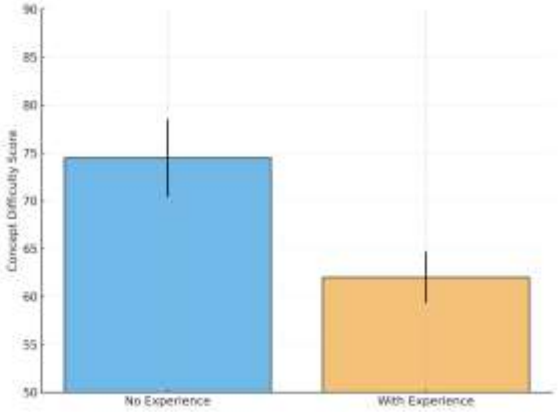

Figure 5. Mean concept difficulty scores for students with and without prior programming experience.

Students with high digital literacy tended to experience lower levels of difficulty. Conversely, those who were complete beginners in programming experienced significantly higher levels of difficulty

Discussion

When the five indicators are analyzed in an integrated manner, it is clear that Physics Education students' readiness to participate in computational learning still requires strengthening in several aspects. Some reflections to draw: Digital skills are no guarantee of programming mastery. Students who are somewhat proficient in operating digital devices do not necessarily quickly grasp program structure. Programming experience remains a differentiating factor. Computational mathematics plays a significant role. The entire algorithmic structure of physics relies on mathematical logic. Students with low mathematical abilities will experience multiple obstacles. Devices are a technical factor that has a psychological impact. Students with slow devices tend to lose motivation when programs don't run. Difficulty with computational concepts is a normal phenomenon. For students encountering code for the first time, it takes time to grasp computational thinking. Lecturers must start from the most basic level. Computing material cannot be delivered instantly. Learning stages must be structured using a gradual, contextual approach, and not be overly stressful.

Overall, the results of this study emphasize the need for more adaptive computing learning strategies, strengthening digital literacy, mathematical skills, and fundamental programming experience. Lecturers need to design approaches that emphasize not only the final program outcomes but also the students' learning journey in understanding computational logic. A gradual and collaborative approach has the potential to help students build confidence, so that the process of learning computing is no longer perceived as a daunting task but as a skill that can be learned progressively. These findings are expected to inform instructors and curriculum developers in designing computing courses that are more inclusive, responsive, and tailored to the abilities of today's physics education students.

CONCLUSION

This study provides a fairly clear picture of the computational readiness of Physics Education students to participate in modern computing learning based on digital technology. Overall, students fall into the moderate readiness category, with several aspects standing out as determining factors. Digital literacy, for example, is a crucial foundation that helps students understand software and computing environments. However, this capability is not evenly distributed, resulting in some students still facing difficulties in the initial stages of using computing applications. Computational mathematics skills also play a crucial role. Students with a strong mathematical foundation tend to understand algorithmic structures more easily, while those with a weak mathematical foundation often struggle to interpret logical operations within code. This directly impacts their adaptability to programming. Programming experience is a significant differentiating factor. Those who have previously experienced programming appear more confident when it comes to understanding code examples or debugging. Conversely, most students who have never interacted with programming require more time to develop a basic understanding. This difference suggests the need for a more gradual and comprehensive teaching strategy. Device readiness also influences students' learning experiences. Low-performance devices often cause frustration and hinder learning, especially when students need to run computing applications. This highlights the importance of technical factors in technology-based learning. The relatively high difficulty level of computing concepts indicates that students require more support in the early stages. Understanding pseudocode, branching logic, looping, and debugging should be provided gradually, with examples relevant to their experiences.

ACKNOWLEDGMENT

The authors would like to express their sincere appreciation to the Coordinator of the Physics Education Study Program for the continuous support, administrative facilitation, and academic guidance provided throughout the implementation of this study. The authors also extend their gratitude to the Physics Education students who willingly participated in the data collection process and contributed valuable insights that made this research possible. Their cooperation and enthusiasm were essential in completing this work.

REFERENCES

-, A. (2023). Transforming educational dynamics: Machine learning's influence on teacher-student interactions in hybrid learning environment. International Journal for Multidisciplinary Research, 5(5). https://doi.org/10.36948/ijfmr.2023.v05i05.8104

Aly, A., & Efendi, E. (2024). Journey of language instruction: An autoethnographic study on teaching English to young learners in Indonesian home-schooling environments. Child Education Journal, 6(2). https://doi.org/10.33086/cej.v6i2.6075

Awuor, F., & Okono, E. (2022). ICT integration in learning of physics in secondary schools in Kenya: Systematic literature review. Open Journal of Social Sciences, 10(09), 421–461. https://doi.org/10.4236/jss.2022.109027

Bitzenbauer, P., Faletič, S., Michelini, M., Tóth, K., & Pospiech, G. (2024). Design and evaluation of a questionnaire to assess learners' understanding of quantum measurement in different two-state contexts: The context matters. Physical Review Physics Education Research, 20(2). https://doi.org/10.1103/physrevphyseducres.20.020136

Connolly, C., Hijón-Neira, R., & Grádaigh, S. (2021). Mobile learning to support computational thinking in initial teacher education. International Journal of Mobile and Blended Learning, 13(1), 1–15. https://doi.org/10.4018/ijmbl.2021010104

Deák, C., & Kumar, B. (2024). A systematic review of STEAM education's role in nurturing digital competencies for sustainable innovations. Education Sciences, 14(3), 226. https://doi.org/10.3390/educsci14030226

Fauzi, A. (2017). Integrating numerical computation into the undergraduate education physics curriculum using spreadsheet Excel. Journal of Physics: Conference Series, 909, 012056. https://doi.org/10.1088/1742-6596/909/1/012056

Fracchiolla, C., & Meehan, M. (2021). Computational practices in introductory science courses. Proceedings of the Physics Education Research Conference, 129–134. https://doi.org/10.1119/perc.2021.pr.fracchiolla

Gawade, V., Bifulco, C., & Guo, W. (2022). Lessons learned to effectively teach and evaluate undergraduate engineers in work design and ergonomics laboratory from a world before, during, and after COVID-19. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 66(1), 756–760. https://doi.org/10.1177/1071181322661505

Handito, T., Mulyono, H., & Khuluqo, I. (2025). Teaching factory and computational thinking in Asian vocational education: A bibliometric analysis of the Scopus database 2015–2025. https://doi.org/10.21203/rs.3.rs-7901437/v1

Hermawan, I., Arjaya, I., & Diarta, I. (2022). Be-raise: A blended-learning model based on Balinese local culture to enhance students' environmental literacy. Jurnal Pendidikan IPA Indonesia, 11(4), 552–566. https://doi.org/10.15294/jpii.v11i4.39475
https://doi.org/10.13189/ujer.2020.080904

Kemouss, H., & Khaldi, M. (2025). Physics teaching with artificial intelligence (AI): A personalized approach for accommodator-style learners according to Kolb. International Journal of Instruction, 18(3), 39–58. https://doi.org/10.29333/iji.2025.1833a

Khan, N., Khan, A., & Rajeyyagari, S. (2025). Innovation in teaching and learning with the use of modern computational tools: A post-COVID experience. Middle East Journal of Applied Science & Technology, 8(2), 74–82. https://doi.org/10.46431/mejast.2025.8208

Mahligawati, F., Allanas, E., Butarbutar, M., & Nordin, N. (2023). Artificial intelligence in physics education: A comprehensive literature review. Journal of Physics: Conference Series, 2596(1), 012080. https://doi.org/10.1088/1742-6596/2596/1/012080

Mercado, J. (2021). Resilient mechanism in learning modern physics: A grounded theory. https://doi.org/10.21203/rs.3.rs-690636/v1

Nkweke, O. (2020). Repositioning physics education through educational information and engineering technology in Bayelsa State, Nigeria. TP, 1, 22–33. https://doi.org/10.71016/tp/6x816v98

Odden, T., & Caballero, M. (2023). Physics computational literacy: What, why, and how? 19–1–19–28. https://doi.org/10.1063/9780735425477_019

Pavlenko, M., Павленко, Л., Iotov, Y., & Pavlenko, Y. (2025). Experience in the development and implementation of the course “Python for Physics Teachers.” Journal of Physics: Conference Series, 3105(1), 012011. https://doi.org/10.1088/1742-6596/3105/1/012011

Quang, D. (2025). From generalization to personalization: Designing a personalized ESP teaching model in a Vietnamese private university context. Social Science and Humanities Journal, 9(04), 7498–7510. https://doi.org/10.18535/sshj.v9i04.1784

Redish, E. (1994). Implications of cognitive studies for teaching physics. American Journal of Physics, 62(9), 796–803. https://doi.org/10.1119/1.17461

Rivadeneira, D., & Toledo, J. (2024). Exploring the impact of computational thinking on teacher education: A systematic review. Revista de Gestão Social e Ambiental, 18(7), e07535. https://doi.org/10.24857/rgsa.v18n7-123

Robeva, R., & Laubenbacher, R. (2009). Mathematical biology education: Beyond calculus. Science, 325(5940), 542–543. https://doi.org/10.1126/science.1176016

Suson, R., Baratbate, C., Anoos, W., Ermac, E., Aranas, A., Malabago, N., … & Capuyan, D. (2020). Differentiated instruction for basic reading comprehension in Philippine settings. Universal Journal of Educational Research, 8(9), 3814–3824.

Travero, A. (2022). Integrating workbook-making in learning calculus during the pandemic: A phenomenological study. International Journal of Studies in Education and Science, 4(1), 19–30. https://doi.org/10.46328/ijses.40

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. TechTrends, 60(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. (2014). Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education, 14(1), 1–16. https://doi.org/10.1145/2576872

Zhang, S., Wang, X., Ma, Y., & Wang, D. (2023). An adaptive learning method based on knowledge graph. Frontiers in Educational Research, 6(6). https://doi.org/10.25236/fer.2023.060624

Ziatdinov, R., & Valles, J. (2022). Synthesis of modeling, visualization, and programming in GeoGebra as an effective approach for teaching and learning STEM topics. Mathematics, 10(3), 398. https://doi.org/10.3390/math10030398