International Journal of Research in Education

Volume 4, Issue 1, January 2024, pp. 140 - 148

e-ISSN: 2745-3553

DOI: https://doi.org/10.26877/ijre.v4i1.17487

Simplifying the Puzzle: How Computational Thinking and Abstraction can Help Teachers Conquer Classroom Complexity

Moses Adeleke Adeoye*[1], Entika Fani Prastikawati[2]

[1*]Al-Hikmah University Ilorin, Nigeria

[2]Universitas PGRI Semarang, Indonesia

*Corresponding Author E-mail: princeadelekm@gmail.com

ARTICLE INFO

Received: November 22, 2023

Revised: January 05, 2024

Accepted: January 10, 2024

Keywords: *Computational Thinking, Abstraction, Classroom Complexity, Problem-Solving, Cognitive Skills*

ABSTRACT

The investigation examines the utilization of abstraction and computational thinking by educators to effectively navigate the complexities encountered in the educational setting. It underscores the significance of these cognitive capacities in the examination and determination of intricate issues. To facilitate students in problem-solving and decision-making, educators can utilize computational thinking to break down challenging issues into more manageable components. Two capacities that educators can utilize to apply their knowledge in various circumstances are the identification of patterns and the extrapolation of abstract notions. This research study demonstrated the utility of abstraction and computational thinking in equipping educators with the necessary tools to address classroom issues, which is particularly valuable for curricula focused on training future educators. The study not only identifies potential challenges but also offers recommendations for overcoming the difficulties that teachers may encounter when implementing these ideas in the classroom. By employing these tactics and recommended solutions, educators can help students improve their analytical, problem-solving, and critical thinking abilities, thus preparing them for the challenges of the digital era.

Introduction

The concepts of abstraction and computational thinking are very useful for teachers to know when navigating the complexity of the classroom. Teachers can improve students' critical thinking, problem-solving, and general involvement by implementing these ideas into the classroom. Computational thinking is a method of problem-solving that divides difficult issues into smaller, more doable components (Alegre et al., 2020; Shailaja & Sridaran, 2015). Effective problem analysis and solving are achieved by utilizing computer science components like data analysis, algorithms, and logic. Students that employ computational thinking get the ability to approach problems methodically and logically which empowers them to come up with original answers (Sung et al., 2017; Yokuş & Kahramanoglu, 2020). Simplifying complicated issues through abstraction entails concentrating on pertinent aspects and disregarding unimportant details (Ezeamuzie et al., 2022). By dissecting difficult ideas into smaller, more digestible parts, it helps learners grasp them. Teachers can assist students in understanding difficult concepts and gaining a deeper comprehension of the subject matter by using abstraction.

The four fundamental components of computational thinking are as follows: Decomposition is the process of breaking down huge, complex problems into smaller, more manageable subproblems. By dissecting a problem into manageable parts, educators may better identify the underlying problems and offer focused answers. The goal of the pattern recognition phase is to identify patterns and similarities within an issue. Instructors can spot trends and offer helpful solutions by drawing on their own experiences and areas of expertise. Abstracting a problem entails removing any details that aren't essential in understanding its main elements. Teachers can better recognise a problem's fundamental components and provide useful solutions by simplifying it. The creation of an algorithm entails organizing a series of directives or actions that must be performed to resolve an issue. Using algorithms, teachers can clearly and rationally demonstrate a way to solve a problem.

Teachers can more adeptly negotiate the intricacies of the classroom by employing these fundamentals of computational thinking (Shailaja & Sridaran, 2015; Shute et al., 2017). They can simplify difficult issues into smaller, more manageable chunks, spot trends and parallels, omit superfluous details, and create efficient, step-by-step processes for problem-solving. Teachers can gain a lot by integrating abstraction and computational thinking into the classroom. Firstly, by encouraging instructors to approach problem-solving in a more organised and methodical manner, these ideas help them better assess and handle the challenges that students face. Second, when learners learn to reason logically and assess different methods of problem-solving, computational thinking and abstraction foster critical thinking abilities in them. As a result, the learning environment becomes livelier and more interesting.

Additionally, students' teamwork and communication abilities are improved through computational thinking and abstraction. Through collaboratively dissecting intricate issues and examining many viewpoints, learners acquire proficient communication and cooperation abilities. These are essential life skills that apply to a variety of personal and professional contexts. Computational thinking and abstraction can significantly influence students' educational experiences (Alegre et al., 2020; Csizmadia et al., 2015). With the help of these ideas, students can solve problems in an organized manner and approach challenges with confidence and creativity. By decomposing intricate problems into smaller, more digestible components, students may comprehend and solve difficulties more efficiently.

Moreover, abstraction and computational thinking provide a deeper comprehension of the subject (Dagienė et al., 2019; Ezeamuzie et al., 2022). Students can understand key concepts and establish a strong foundation of knowledge by breaking down difficult concepts into simpler terms. They perform better academically and are better prepared for future learning opportunities. By using this strategy, educators can more successfully deconstruct difficult subjects into manageable parts, increasing students' access to and engagement with the material. This improves students' ability to solve problems and helps them gain a deeper comprehension of the material.

Abstracting ideas is one technique to apply computational thinking. Seeing the important aspects of a problem and disregarding the elements that are not relevant is the process of abstraction. Teachers can concentrate on the essential ideas and help students grasp them better by simplifying difficult subjects. For instance, when instructing students on a difficult mathematical formula, teachers might simplify the fundamental ideas and offer practical examples to demonstrate how they are applied. Algorithmic thinking is another facet of computational thinking. This calls for the capacity to create methodical processes or algorithms for problem-solving. Teachers can assist students in approaching complex topics methodically and logically by teaching them how to think algorithmically (Ezeamuzie et al., 2022; Damelin et al., 2019). This streamlines the educational process and gives learners transferable problem-solving abilities across multiple topic areas.

The significance of pattern recognition is further emphasized by computational thinking (Bilbao et al., 2017). Teachers can assist students in drawing connections and gaining valuable insights by pointing out patterns in data or information. For example, teachers can assist students in identifying patterns in cause and effect when they analyze historical events, which can help them get a deeper understanding of the subject. Moreover, iterative design and testing are fostered by computational thinking. To improve the answer, this entails segmenting a problem into smaller subproblems, building prototypes, and testing them. Teachers can help students develop creativity and resilience in the face of difficult problems by pushing them to refine and iterate on their answers.

Metaphors and analogies are two methods in which educators might include abstraction into their lessons. Teachers can build a bridge from the unknown to the familiar by drawing links between difficult ideas and well-known subjects. This improves the content knowledge of the students and makes it easier for them to comprehend challenging ideas. Dividing difficult subjects into manageable, digestible portions is an additional tactic. The most crucial concepts should be identified, and then rationally arranged, to accomplish this. By breaking concepts down into digestible steps and ensuring that students comprehend each one before moving on to the next, teachers may aid students in learning.

Difficult issues can also be artistically illustrated through abstraction. Visual aids such as charts, graphs, and diagrams can be utilized to assist students in making connections between different ideas and visualize complex topics. This visual tool promotes participation and active engagement in the learning process in addition to improving comprehension (Repenning et al., 2016). Additionally, computational thinking can be a helpful strategy for demystifying complex concepts (Swaid, 2015). This approach breaks down problems into smaller, easier-to-manage parts, and then builds algorithms to handle those parts. By applying computational thinking abilities to analyze and identify patterns and relationships, teachers may simplify challenging subjects. One of the core concepts of computational thinking, decomposition, can be used by educators to break down a challenging subject into its constituent parts. Following that, they can focus on teaching each element independently before putting them together to produce a thorough understanding of the subject. Furthermore, abstraction can help teachers develop effective procedures for assessment and evaluation. By identifying the key concepts and competencies that students must possess, educators may design assessments that measure understanding and application rather than only memorization. Teachers who break down complex concepts into smaller, more accessible pieces will find that their students comprehend and are more engaged in difficult subjects.

Research Methods

This study employed a systematic literature review methodology. Researchers collected journal articles from Google Scholar, Research Gate, SINTA and Web of Science related to computational thinking, abstraction and their application in the classroom. To gain a deep understanding of the existing knowledge and research in the field, identifying gaps and areas were further investigated.

Findings

As the educational panorama continues to evolve, the mixing of computational wondering and abstraction in coaching has emerged as an effective tool for educators to overcome the complexities of the modern-day study room. In our examination, we discovered that incorporating computational wondering and abstraction not handiest complements college students' trouble-fixing talents. however, additionally fosters a deeper expertise of complex concepts throughout various topics. To correctly enforce these strategies in the lecture room, educators can hire a variety of effective techniques that cater to numerous gaining knowledge of styles and sell critical thinking talents.

Incorporating Computational Thinking and Abstraction in Teaching

To manage complexity in the classroom, the ideas of abstraction and computational thinking are extremely relevant (Tran, 2019; Rich et al., 2019; Dagienė et al., 2019; Sung et al., 2017). Teachers can improve students' comprehension of difficult subjects and streamline the learning process by implementing these approaches. Teachers can determine the essential elements of an issue and provide effective solutions by applying computational thinking. With this method, educators can break down complex subjects and make them more understandable for the students. Abstractionism in education is the process of reducing complicated topics to simpler, easier-to-understand principles. Teachers can provide students with a clear and simple knowledge of the subject matter by abstracting complex facts (Weintrop et al., 2016). By keeping things simple, learners can understand the essential ideas without being bogged down in extraneous details.

Instructors may effectively manage complexity in the classroom when they incorporate abstraction and computational thinking into their lesson plans. By decomposing complicated problems into manageable chunks, educators can pinpoint the precise areas in which students might find it difficult. This enables them to offer interventions and focused support to meet each student's unique learning needs. Furthermore, abstraction and computational thinking support educators in designing organized learning activities. Teachers can create classes that are more interesting and understandable for students by breaking down difficult ideas into simpler terms and concentrating on key ideas. This method fosters critical thinking and problem-solving skills in children while also encouraging active learning.

Moreover, abstraction and computational thinking promote the growth of higher-order cognitive abilities. Teachers give students the tools they need to analyze and solve real-world situations by teaching them how to abstract complicated material and break down problems. These qualities can help students prepare for future occupations that need the ability to solve problems and think critically. They are also transferable to other academic areas.

Strategies for Implementing Computational Thinking and Abstraction in the Classroom

Designing lessons with abstraction and computational thinking in mind is essential. Students who possess this talent can take significant concepts and patterns from real-world scenarios and convert them into a more palatable style. Teachers can foster critical thinking, pattern recognition, and a deeper comprehension of the material by integrating abstraction into their lesson plans. Students can gain from lesson design that incorporates abstraction and computational thinking in several ways. First, practice reasoning and problem-solving skills. Through computational thinking tasks, students develop their ability to approach problems methodically, consider multiple points of view, and construct algorithms to solve them. This helps pupils become more proficient in mathematics and logical reasoning and prepares them to overcome challenges in a range of disciplines (Rich et al., 2019; Sung et al., 2017; Wing, 2008). Firstly, practice reasoning and problem-solving skills. Through computational thinking tasks, students develop their ability to approach problems methodically, consider multiple points of view, and construct algorithms to solve them. This helps pupils become more proficient in mathematics and logical reasoning and prepares them to overcome challenges in a range of disciplines (Rich et al., 2019; Sung et al., 2017; Wing, 2008). Secondly, combining abstraction and computational thinking fosters innovation and creativity. Teachers can cultivate a mindset that favours unconventional solutions and out-of-the-box thinking by pushing learners to break down complex problems and think abstractly (Shute et al., 2017). This could lead to the development of original ideas and solutions, which are highly prized in today's environment which is changing rapidly.

Including these ideas in lesson design can also improve students' computational literacy. To navigate technology and make wise decisions in an increasingly digital world, one must grasp the concepts of computational thinking and abstraction (Yokuş & Kahramanoglu, 2022). Teachers may equip students to take an active role in the rapidly expanding digital society by teaching them these abilities. Teachers can use a variety of tactics to include abstraction and computational thinking in lesson design. First of all, they can create lessons that include real-world issues and motivate students to use computational thinking methods to find solutions. Coding exercises, simulations, and hands-on activities can all be used to do this. Teachers can also allow students to exercise abstraction by breaking down difficult ideas or systems into simpler forms. They can help students recognize patterns, build models, and convey information in a more comprehensible and succinct manner. Concept maps, diagrams, and visualizations can be used to accomplish this.

Discussion

Benefits of Computational Thinking and Abstraction in Education

Learning results for students are positively impacted by the many advantages of computational thinking and abstraction. Students acquire useful abilities that are transferable to different academic fields and real-world situations by integrating these ideas throughout their education. Its capacity to improve problem-solving abilities is one of computational thinking's primary benefits. Learners learn how to deconstruct complicated issues into smaller, easier-to-manage parts. Through in-depth analysis and comprehension of the issue, students can come up with workable solutions. This method promotes logical reasoning and critical thinking, two abilities that are necessary for success in both academic and professional contexts.

By abstracting information, students can make connections and apply their knowledge to new contexts. This transferable skill is essential in today's world of rapid change when the capacity to adapt and apply knowledge is crucial. Furthermore, creativity and invention are stimulated by abstraction and computational thinking. By encouraging students to think creatively and approach problems from multiple angles, these concepts promote an experimental and exploratory mindset. By abstracting information, students can make connections and apply their knowledge to new contexts. This transferable skill is essential in today's world of rapid change when the capacity to adapt and apply knowledge is crucial. Furthermore, creativity and invention are stimulated by abstraction and computational thinking. By encouraging students to think creatively and approach problems from multiple angles, these concepts promote an experimental and exploratory mindset. Furthermore, creativity and invention are stimulated by abstraction and computational thinking. By encouraging students to think creatively and approach problems from multiple angles, these concepts promote an experimental and exploratory mindset. Students are encouraged to brainstorm several options and assess each one's efficacy. This strategy helps them approach problems more imaginatively, take measured risks, and learn from their failures. Another benefit of computational thinking and abstraction is that it improves learners' ability to reason algorithmically. Algorithmic thinking is the process of formulating a solution to a problem using a series of sequential instructions. Students who are proficient in this skill approach problem-solving in a more planned and logical manner. They construct logical sequences of events that aid in the resolution of computational problems and improve their overall problem-solving abilities.

Additionally, abstraction and computational thinking support multidisciplinary education (Yokuş & Kahramanoglu, 2022; Selby & Woollard, 2014). These ideas cut across conventional disciplinary boundaries, enabling students to use what they've learned in a variety of settings. Computational thinking and abstraction offer a common vocabulary and structure for problem-solving, whether one is solving mathematical puzzles, evaluating scientific data, or creating computer programs (Swaid, 2015). This multidisciplinary method promotes teamwork and pushes students to consider issues from several angles.

Using Computational Thinking and Abstraction: Challenges and Solutions

When implementing abstraction and computational thinking in the classroom, educators may encounter several challenges. Nevertheless, by taking the right steps, these challenges can be effectively overcome. One major barrier would be educators' lack of training and experience with computational thinking and abstraction. Many teachers may be untrained or are not familiar with these concepts. Schools and other educational establishments can help tackle this problem by offering their teachers opportunities for professional development. Educators' understanding and skill in computational thinking and abstraction could be enhanced through seminars, training sessions, and web-based resources. By funding teachers' professional development, schools may ensure that they possess the knowledge and confidence necessary to implement these concepts successfully. Getting the appropriate instruments and materials is another issue. Finding curriculum-aligned materials and tools that promote computational thinking and abstraction may prove challenging for teachers. To get over this challenge, schools might collaborate with nearby corporations, charitable organisations, and IT firms to obtain grants or donations for teaching resources. Schools may also collaborate with universities or technology centres to obtain access to state-of-the-art tools and resources. By providing educators with the necessary resources, schools may facilitate the incorporation of abstraction and computational thinking into the classroom.

Moreover, incorporating abstraction and computational thinking into current curricula can be difficult. It could be challenging for teachers to incorporate these ideas into their lesson plans and teaching methods. Teachers, administrators, and curriculum specialists might form a committee or team in schools to build curricula to address this issue. Collaboratively, this team may create policies, frameworks, and lesson plans that integrate computational thinking and abstraction in a variety of topics and academic levels. Schools can facilitate the integration of these principles into the teaching process by offering teachers pre-made tools. Furthermore, evaluating and assessing computational thinking and abstraction can be difficult. The comprehension and application of these concepts by learners may not be adequately measured by traditional assessment techniques. Alternative assessment techniques, including project-based exams, coding competitions, or group problem-solving exercises, can help schools get past this obstacle. With the use of these techniques, students' abilities in computational thinking and abstraction can be more thoroughly and authentically assessed. Instructors might also include components for peer and self-assessment to promote students' introspection and feedback.

Conclusion

The findings of this study highlight how important abstraction and computational thinking are in helping teachers overcome obstacles in the classroom. By implementing these strategies, teachers can facilitate learning, enhance students' capacity for problem-solving, and promote critical thinking. The research highlights the significance of integrating computational thinking techniques into the curriculum to adequately equip students for the digital age. It also emphasizes how important it is for educators to continue their professional development to apply these tactics in the classroom. Teachers can foster a more inclusive and engaging learning environment that gives students the tools they need to succeed in an increasingly technologically-dependent society by embracing computational thinking and abstraction.

References

Alegre, F., Underwoood, J., Moreno, J. & Alegre, M. (2020, February). Introduction to computational thinking: a new high school curriculum using code world. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 992-998).

Bell, T. & Lodi, M. (2019). Constructing computational thinking without using computers. *Constructivist Foundations, 14*(3), 342-351.

Bilbao, J., Bravo, E., García, O., Varela, C. & Rebollar, C. (2017). Assessment of computational thinking notions in secondary school. Baltic Journal of Modern Computing, 5(4), 391 -397.

Chalmers, C. (2018). Robotics and computational thinking in primary school. International Journal of Child-Computer Interaction, 17, 93-100.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking guide for teachers. https://community.computingatschool.org.uk/resourcesl2324/single

Czerkawski, B. (2013, March). Instructional design for computational thinking. In Society for Information Technology & Teacher Education International Conference (pp. 10-17). Association for the Advancement of Computing in Education (AACE).

Dagienė, V., Futschek, G., & Stupurienė, G. (2019). Creativity in solving short tasks for learning computational thinking. Constructivist Foundations, 14(3), 382-396.

Damelin, D., Stephens, L. & Shin, N. (2019). Engaging in computational thinking through system modelling. Concord, 23(2), 4-6.

Ezeamuzie, N.O., Leung, J.S. & Ting, F.S. (2022). Unleashing the potential of abstraction from cloud of computational thinking: A systematic review of literature. Journal of Educational Computing Research, 60(4), 877-905.

Kalelioglu, F., Gülbahar, Y. & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. Baltic Journal of Modern Computing, 4(3), 583.

Kong, S.C. (2016). A framework of curriculum design for computational thinking development in K-12 education. Journal of Computers in Education, 3(4), 377-394.

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L. & Settle, A. (2014, June). Computational thinking in K-9 education. In Proceedings of the working group reports of the 2014 innovation & technology in computer science education conference (pp.1-29).

Marcelino, M.J., Pessoa, T., Vieira, C., Salvador, T. & Mendes, A.J. (2018). Learning computational thinking and scratch at distance. Computers in Human Behavior, 80, 470-477.

Repenning, A., Basawapatna, A. & Escherle, N. (2016, September). Computational thinking tools. In 2016 IEEE symposium on visual languages and human-centric computing (VL/HCC) (pp. 218-222). IEEE.

Rich, K.M., Yadav, A. & Schwarz, C.V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. Journal of Technology and Teacher Education, 27(2), 165-205.

Selby, C. & Woollard, J. (2014). Refining an understanding of computational thinking.

Settle, A., Franke, B., Hansen, R., Spaltro, F., Jurisson, C., Rennert-May, C. & Wildeman, B. (2012, July). Infusing computational thinking into the middle and high-school curriculum. In Proceedings of the 17th ACM annual conference on Innovation and Technology in Computer Science Education (pp. 22-27).

Shailaja, J. & Sridaran, R. (2015). Computational thinking the intellectual thinking for the 21st century. International Journal of Advanced Networking & Applications, 7, 39-46.

Shute, V.J., Sun, C. & Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational research review, 22, 142-158.

Sung, W., Ahn, J. & Black, J.B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. Technology, Knowledge and Learning, 22, 443-463.

Swaid, S.I. (2015). Bringing computational thinking to STEM education. Procedia Manufacturing, 3, 3657-3662.

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. Journal of Educational Computing Research, 57(1), 3-31.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147.

Wing, J.M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Wing, J.M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 366(1881), 3717-3725.

Yokuş, E. & Kahramanoglu, R. (2022). An Overview of Computational Thinking. Anemon Muş Alparslan Üniversitesi Sosyal Bilimler Dergisi, 10(1), 157-173.