EMPOWERING LEARNERS FOR THE FUTURE: INTEGRATING COMPUTATIONAL THINKING IN SCIENCE AND MATHEMATICS EDUCATION

Rejaul Karim Barbhuiya
Central Institute of Educational Technology, NCERT
Email: rejaul.ncert@nic.in

The rapid permeation of technology in our society demands the integration of Computational Thinking (CT) skills into education to empower learners to navigate technological advancements, drive innovation and prepare for the future workforce. CT is crucial for problem-solving, data analysis and decision-making across domains and its inclusion in science and mathematics education offers numerous benefits. This research paper examines the significance of integrating CT skills in school curricula, focusing on science and mathematics. It explores the elements of computational thinking and their relevance in nurturing analytical and critical thinking abilities among students. The paper also discusses challenges and opportunities in implementing CT education and proposes pedagogical approaches for effective integration. The relationship between computational thinking, science and mathematics is highlighted, emphasising the mutual benefits of this integration for deeper understanding and interdisciplinary connections. Overall, the paper emphasises equipping students with CT skills to prepare them for the ever-evolving technological landscape and foster creativity and problem-solving abilities.

Keywords: Computational Thinking, Problem Solving, 21st Century Skill, Algorithmic Thinking, Decomposition, Pattern Recognition

Introduction

Technology is permeating our society at an ever-increasing pace and touching more and more aspects of our life. Computational Thinking (CT) skills empower learners to navigate technological advancements, reap the benefits of these advancements and drive innovation across various fields. As Artificial Intelligence (AI) and Machine Learning (ML) continue to advance, it becomes imperative for young learners to have CT skills to understand and collaborate with AI technologies. This will empower them to adapt and thrive in the future workforce rather than being replaced by technological advancements.

In the world of rapid automation and innovation, we need a resilient workforce capable of addressing complex problems and challenges faced by societies. CT skills are considered to be crucial for problem-solving, data analysis and data-driven decision-making across domains (Barr, Harrison and Conery, 2011). Different industries—services, transportation, finance, entertainment, education, healthcare, etc., are increasingly deploying data-driven methodologies, AI techniques and other automation. People equipped with CT skills can employ the concepts such as decomposition, abstraction, algorithm and pattern recognition to drive innovation, improve efficiency and tackle complex problems systematically (Riley and Hunt, 2014).

The National Education Policy (NEP, 2020) emphasises inclusion of mathematical and CT throughout the school education, starting from the foundational stage. CT skills can help students learn to analyse problems, evaluate all possible solutions and make informed decisions based on evidence and data. Students can enhance their analytical and critical thinking abilities by honing their ability to break down problems into smaller manageable components and write clearly defined step-wise solutions. This can be useful to nurture creativity and innovation, empowering students to devise novel solutions for computer-related tasks and in other real-world situations where logical reasoning is required.

Integrating CT into the science and mathematics curriculum right from the school stage offers a holistic approach to fostering the essential skills among students (Weintrop et al., 2016). It allows them to gain confidence and expertise in using various software and tools, which are important for understanding the concepts in science and mathematics. CT augments mathematical understanding by providing students with practical contexts to apply mathematical concepts. The integration of coding, algorithm and programming concepts within mathematics lessons can help in having a more profound comprehension of abstract mathematical concepts. Besides developing CT skills early, one can lay a foundation for students interested in pursuing careers in computer science and related fields.

This article inquires into the background and significance of integrating CT skills in school curricula, focusing on science and mathematics subjects. While emphasising the pedagogical benefits of incorporating this 21st-century skill into the educational framework, it discusses a comprehensive integration strategy considering the curricular load and age appropriateness for the young learners.

Computational Thinking

Computational Thinking, or CT, is a cognitive process borrowed from computer science. It is the ability to formulate and solve any problem effectively using computational tools. Jeanett Wing coined the term in her seminar paper and since then, it has been considered one of the crucial skills for the 21st century (Wing, 2006). CT encompasses using fundamental computational concepts, such as algorithms, abstraction, decomposition and pattern recognition, to analyse and solve real-world problems systematically.

On a technical note, CT is the thought process of formulating a problem and expressing its solution in a way that either a human or a machine can effectively carry out the solution (Wing, 2006). CT involves thinking like a computer scientist, but is not limited to coding, programming or thinking like a computer. It is about designing a solution using automation but without necessarily using any computing technology. The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association have defined CT in the following terms (ISTE and CSTA, 2011):

- Express a problem in a way that a computer or a human can solve it. However, the problem-solving formulation does not necessarily require the use of a computer (Grover and Pea, 2018).
- Devise probable solutions and articulate them as algorithms (a sequence of ordered steps).
- Arrange data logically for analysis using various tools and techniques.

- Present data through abstractions such as models and simulations.
- Evaluate multiple solution paths, and identify and implement the most efficient and effective one. The implementation may or may not require the use of a computer.
- Generalise and share this problem-solving process to enable others apply it to a range of problems.

Elements of CT

CT requires a range of competencies that computer scientists frequently utilise to tackle complex problems. The fundamental concepts of computer science used in CT encompass the following:

1. **Logical Thinking:** This involves analysing facts or situations to arrive at a decision or a conclusion. Boolean logic is used in designing electronic circuits and writing the flow of control in programmes. Logic is fundamental to the Artificial Intelligence, allowing AI systems to represent and organise knowledge. Knowledge about a domain or the world can be expressed through logical statements such as propositions, rules and facts, which AI systems use to make inferences and draw conclusions. Through deductive reasoning, AI systems can derive new information based on already known facts and rules.

   As part of CT competency, students can develop logical thinking by solving logical puzzles and constructing boolean expressions using formal boolean logic involving boolean operators such as AND, OR, NOT, etc. (Grover and Pea, 2018).

2. **Algorithmic Thinking:** An algorithm is a step-by-step set of instructions or procedures designed to solve a specific problem or accomplish a particular task. It involves breaking down complex problems into smaller, manageable steps and designing a systematic sequence of instructions or algorithms to solve them. Examples of algorithms include precise steps to follow a recipe for cooking, and GPS-enabled map software that calculates the best route considering factors like traffic and road advisories. These algorithms ensure a systematic, unambiguous and logical sequence of instructions to achieve the desired outcomes.

   Algorithmic thinking is the skill needed for developing an algorithm. Computer scientists write algorithms so that the precise solution steps or instructions can be coded in a programming language. An algorithm can be represented as a flowchart or using pseudocodes. As part of CT competency, students can learn about the three basic building blocks of an algorithm—sequence, selection and repetition. Steps in an algorithm are typically followed sequentially, but conditional checks allow for skipping or repeating certain steps. If-then-else constructs represent conditions, and do-while, for, and repeat constructs enable repetition.

3. **Decomposition:** Human memory is limited and real-world problems often pose challenges too complex to solve without breaking them down into

smaller subproblems. The process known as decomposition helps with this issue as it involves analysing the problem as a whole and dividing it into manageable steps for processing in the brain (IEA, 2016, p. 1) or using a computer. Successfully decomposing a problem requires understanding what constitutes manageable steps and how they relate to each other and the overall problem. Once decomposed, each subproblem should be independent of the others, that is they can be solved or processed in parallel. When their solutions are combined, we obtain the solution to a larger problem. Decomposition makes problems more approachable and simplifies the problem-solving process.

The inclusion of decomposition in the school curriculum cultivates systematic thinking, inspiring students to structure their thoughts and actions logically. The concept is relevant across subjects and disciplines, enriching students' understanding of science, mathematics, technology and other academic areas.

4. **Pattern Recognition:** Patterns are abundant in nature and are visible in flowers, leaves, honeycombs, spider webs, animals, weather, etc. Students are familiar with patterns in geometric shapes and number series. Building on this concept, computer scientists developed the idea of generalising solutions that can be reused in similar problems or situations. Repeated patterns or steps are represented using repetition or loop in programming.

   Further, in programming, when a set of instructions are executed with some change in the input values, those are implemented in the form of a function. In problem-solving, pattern recognition helps identify the part of a problem with similarity to an earlier solved problem so that the solution can be applied here. This is called design patterns in the world of software development. Pattern recognition finds application in the domain of image recognition, natural language processing, machine learning, etc.

   Pattern recognition is a fundamental CT skill that involves observing and identifying recurring structures or regularities in data. It allows us to look for similarities and relationships among different problem elements, scenarios or datasets.

5. **Abstraction:** Wing (2006) identifies abstraction as the central and important cognitive process in CT. In basic terms, abstraction involves simplifying intricate problems by focusing on essential aspects while hiding irrelevant and detailed information that are not essential for current problem-solving purposes. This approach allows individuals to concentrate solely on the necessary aspects, avoiding unnecessary complexity. Abstraction, thus, offers a means of managing and simplifying complex situations, making problem-solving more efficient and effective.

   As per Wing (2014), abstraction involves capturing essential properties common to a group of objects while disregarding irrelevant differences. For instance, an algorithm is an abstraction of a process that takes inputs, executes a sequence of steps and produces outputs to achieve a specific objective. Students with abstraction skills demonstrate increased proficiency in understanding and resolving real-world problems across different subjects and disciplines.

Numerous disciplines require, encourage, and teach mathematical thinking, logical thinking, problem-solving skills and algorithmic thinking (Hsu, Chang and Hung, 2018; Ye et al., 2023). But computer science helps understand and implement computational thinking methods to address challenges in various domains and interdisciplinary areas (Barr and Stephenson, 2011).

Integration of CT with Science and Mathematics

A review of previous studies examining the integration of CT at schools suggests that incorporating it in science (Tucker-Raymond, Cassidy and Puttick, 2021; Basu et al., 2016; Sengupta et al., 2013) and mathematics (Guggemos, 2021; Durak and Saritepeci, 2018; Costa, Campos and Guerrero, 2017) yields numerous benefits. Integration of CT concepts, such as algorithmic thinking, pattern recognition and abstraction, enhances students' problem-solving abilities and analytical skills (Tabesh, 2017; Yadav, Hong and Stephenson, 2016; Shute, Sun and Asbell-Clarke, 2017; Riley and Hunt, 2014; Qualls and Sherrell, 2010). Additionally, it promotes creativity, logical reasoning and systematic thinking (Lee et al., 2020; Grover and Pea, 2018).

Students already learned the elements of CT skills in various disciplines, but ensuring they master the entire skill set empowers them to apply these skills in diverse contexts (Barr, Harrison and Conery, 2011). Studies indicate that students who engage in CT activities demonstrate improved performance in science and mathematics (Barcelos et al., 2018). Moreover, integrating CT into the curriculum enhances students' understanding of real-world applications, such as data analysis and modelling (Reichert, Barone and Kist, 2020; Shute, Sun and Asbell-Clarke, 2017; Qualls and Sherrell, 2010). Students' engagement in CT is often found to be high, especially when activities are hands-on, interactive, and involve problem-solving in real-world contexts (Barr, Harrison and Conery, 2011; Durak and Saritepeci, 2018; Guggemos, 2021).

In mathematics, CT plays a critical role in numerical analysis and optimisation. Algorithms help streamline mathematical processes like solving systems of equations and approximating solutions to complex problems (Costa, Campos and Guerrero, 2017). Additionally, CT aids in pattern recognition, a vital skill in mathematics for identifying trends and generalising solutions.

In science, integrating CT enables data-driven exploration and modelling of scientific phenomena. Using computer-based methods including simulations, agent-based models, collection and analysis of large data sets help students explore big, complicated science topics like climate change, public health and evolution. For example, in a school-level science project, students might use computer simulations to explore chemical reactions. To incorporate CT concepts, we could encourage teachers to guide a discussion about how the simulation represents the real-world reactions, what variables and parameters are manipulated in the simulation, and which aspects of the reactions are simplified or abstracted in the simulation.

Studies have shown that harmonious relationships between CT and science lead to the creation of CT-oriented science curricular units designed for K-12 classrooms (Sengupta et al., 2015; Basu et al., 2016; Repenning, Webb and Ioannidou, 2010). Students can use CT skills to analyse experimental data, draw meaningful conclusions and identify patterns in their scientific investigations. They can develop mathematical models of physical systems, such as projectile motion or simple pendulums, that can help them to explore and predict real-world behaviours. Furthermore, CT skills facilitate the exploration of chemical reactions and molecular structures, deepening students' understanding of chemistry concepts.

The relationship between CT, science and mathematics is interconnected and mutually beneficial. As CT enhances problem-solving and analytical skills, it empowers science and mathematics students to explore, analyse and understand the intricacies of the world around them in deeper and more unambiguous ways. Science and mathematics, in turn, offer real-world contexts for applying CT concepts, making the learning experience more practical and enriching for students. The integration of these fields, therefore, can create a new generation of learners who can approach challenges with a multidisciplinary perspective and adapt to the ever-evolving technological landscape.

Nurturing CT skills requires a thoughtful selection of pedagogical approaches that engage students and develop their problem-solving, logical thinking, algorithmic thinking, and analytical abilities. Some of these approaches can be:

- Project-based learning
- Unplugged activities (computational activities without using a computer)
- Game-based learning (including Puzzle, Sudoku)
- Coding and programming

The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) have come up with useful teacher resources on integrating CT across subject areas in K–12 Education (ISTE and CSTA, 2021). There can be three 3 levels of CT integration at a broader level (Waterman *et al.,* 2018):

- At an initial level, there can be discussion about the CT ideas, abilities and ways of doing things that are already present in the lessons. These chances to use CT might not require using computers, but teachers can show how they can also connect to computers or other technology. For instance, students often create or utilise physical models to understand or explain scientific concepts. Teachers can link this to CT by discussing what a model represents, how it conveys the concept and what it includes or excludes.
- At an intermediate level, lessons can be developed to strengthen the core subject understanding and establish

a direct link to computing concepts that are already part of the lesson but not the main focus. For instance, in a regular science lesson, students might conduct a simple experiment to observe the effects of temperature on plant growth. A CT enhancement activity could involve designing an experiment that monitors multiple variables simultaneously, using sensors and data collection software to record and analyse data more efficiently. Additionally, students could employ time-lapsed video recording to capture and analyse the plant growth over an extended period. This way, students not only explore the scientific concept but also gain CT skills related to experimental design, data analysis and multimedia data processing.

- At a higher level, fresh lesson plans or a series of lessons can be developed to expand the core disciplinary concept, while serving as a foundation for exploring computer science. This often includes programming exercises. For instance, students can engage in computer simulations and make adjustments to variables or the underlying code to explore the dynamic changes in systems over time. Taking an example of the concept of chemical reactions, students could conduct simulated experiments with different reactants and observe how the reactions change over time. By modifying variables like temperature or concentration, students gain a deeper understanding of how these dynamic systems behave, all while applying CT principles to design and analyse their experiments.

Challenges and Opportunities in Integrating CT

Implementing an integrated approach to CT in education comes with challenges and opportunities. Some of the imminent challenges are:

1. **Teacher Readiness:** Many educators may not be familiar with CT concepts or lack experience in integrating them into their teaching. Studies (Barr and Stephenson, 2011; Ketelhut et al., 2020) have highlighted the importance of teacher professional development and support in effectively integrating CT into the curriculum. Teachers' knowledge and confidence in using CT approaches play a significant role in promoting students' engagement and success.
2. **Restructuring Curriculum:** The curriculum needs to be restructured in order to infuse CT concepts and problem-solving approaches across various subjects. Coordinating efforts between different subject teachers and finding appropriate points for integration can be challenging (Tucker-Raymond, Cassidy and Puttick, 2021).
3. **Assessment and Evaluation:** Designing assessment methods to evaluate students' computational thinking skills accurately can be challenging. Traditional assessment methods may not fully capture the depth of different CT abilities (Fields et al., 2021).

4. **Time Management:** The existing curriculum and the school schedule may already be packed with other subjects and activities, leaving limited time for integrating computational thinking. Finding the right balance without overburdening students and teachers can be a challenge.

The integration of CT into other subjects presents various opportunities, including but not limited to the following:

- **Enhanced Problem-Solving Abilities:** CT helps develop systematic problem-solving skills which can be applied to solve real-world problems in many domains. This makes students confident in handling complex problems.
- **Interdisciplinary Connections:** Students can realise the interconnections between subjects and apply computational concepts in different contexts. This can help get a deeper understanding of how concepts are interconnected (Alam, 2022), and make them proficient in handling open-ended problems.
- **Collaborative Learning:** Students engage in group projects and develop good communication and collaboration skills as they work together to achieve common goals.
- **Critical and Analytical Thinking:** An integrated approach helps develop critical thinking and analytical abilities as students learn to analyse data, model situations, and make informed decisions based on computational processes. This increases adaptability and tolerance in situations having inherent ambiguities.

Conclusion and Way Forward

The ever-evolving technological landscape necessitates the integration of CT skills into school education. CT empowers individuals to navigate technological advancements, drive innovation and collaborate with artificial intelligence and machine learning technologies. It is essential to equip young learners with CT skills to solve complex real-world problems and adapt to the future. CT is crucial for problem-solving, data analysis and decision-making across various domains and industries. Integrating CT into the science and mathematics curriculum offers a holistic approach to fostering essential skills among students, enhancing their problem-solving abilities, and nurturing creativity and innovation.

The relationship between CT, science, and mathematics is interconnected and mutually beneficial. CT enhances problem-solving abilities, promoting interdisciplinary connections and fostering critical and analytical thinking, enabling science and mathematics students to explore and understand complex concepts more effectively. The integration of these fields creates a multidisciplinary perspective. While integrating CT into education offers numerous benefits, there are challenges to overcome, including teacher readiness, curriculum restructuring and designing appropriate assessment methods.

To successfully integrate CT into education, emphasis should be placed on teacher professional development and continued support to ensure that educators are well-equipped to incorporate CT concepts into their teaching. Collaboration between different subject teachers is essential for finding appropriate points of integration and ensuring a seamless learning experience for students.

Pedagogical approaches like project-based learning, unplugged activities, game-based learning and coding can effectively engage students and develop their problem-solving and analytical abilities. Additionally, innovative assessment methods should be designed to accurately evaluate students' CT skills and capture the depth of their capabilities.

References

Alam, A. 2022. Educational Robotics and Computer Programming in Early Childhood Education: A Conceptual Framework for Assessing Elementary School Students' Computational Thinking for Designing Powerful Educational Scenarios. In 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN). IEEE. pp. 1–7.

Barcelos, T. S., R. Muñoz-Soto, R. Villarroel, E. Merino and I. F. Silveira. 2018. Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review. *Journal of Universal Computer Science*. Vol. 24, No. 7. pp. 815–845.

Barr, D., J. Harrison and L. Conery. 2011. Computational Thinking: A Digital Age Skill for Everyone. *Learning and Leading with Technology*. Vol. 38, No. 6. pp. 20–23.

Barr, V. and C. Stephenson. 2011. Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads*. Vol. 2, No. 1. pp. 48–54.

Basu, S., G. Biswas, P. Sengupta, A. Dickes, J. S. Kinnebrew and D. Clark. 2016. Identifying Middle School Students' Challenges in Computational Thinking-Based Science Learning. *Research and Practice in Technology Enhanced Learning*. Vol. 11, No. 1. pp.1–35.

Costa, E.J.F., L.M.R.S. Campos and D.D.S. Guerrero. 2017. Computational Thinking in Mathematics Education: A Joint Approach to Encourage Problem-Solving Ability. *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE. pp. 1–8.

Durak, H. Y. and M. Saritepeci. 2018. Analysis of the Relation Between Computational Thinking Skills and Various Variables with the Structural Equation Model. *Computers and Education*. Vol. 116. pp. 191–202. https://doi.org/10.1016/j.compedu.2017.09.004

Fields, D., D. Lui, Y. Kafai, G. Jayathirtha, J. Walker and M. Shaw. 2021. Communicating about Computational Thinking: Understanding Affordances of Portfolios for Assessing High School Students' Computational Thinking and Participation Practices. *Computer Science Education*. Vol. 31, No. 2. pp. 224–258.

Grover, S. and R. Pea. 2018. Computational Thinking: A Competency Whose Time Has Come. *Computer Science Education: Perspectives on Teaching and Learning in School.* Vol. 19, No. 1. pp. 19–38.

Guggemos, J. 2021. On the Predictors of Computational Thinking and its Growth at the High-School Level. *Computers and Education*. Vol. 161. 104060.

Hsu, T. C., S. C. Chang and Y. T. Hung. 2018. How to Learn and How to Teach Computational Thinking: Suggestions Based on a Review of the Literature. *Computers and Education.* Vol. 126. pp. 296–310.

IEA. 2016. What's next for IEA's ICILS in 2018? *The IEA's International Computer and Information Literacy Study (ICILS) 2018*. Retrieved on 12 May 2023 from http://www.iea.nl/fileadmin/user_upload/Studies/ICILS_2018/IEA_ICILS_2018_Computational_Thinking_Leaflet.pdf

ISTE and CSTA. 2011. *Computational Thinking in K–12 Education Leadership Toolkit.* ISTE. https://cdn.iste.org/www-root/2020-10/ISTE_CT_Leadership_Toolkit_booklet.pdf

———. 2021. Computational Thinking in K–12 Education - Teacher Resources. Retrieved on 5 May 2023 from https://cdn.iste.org/www-root/2020-10/ISTE_CT_Teacher_Resources_2ed.pdf?_ga=2.228619804.1439222402.1696653720-944240924.1690046886

Ketelhut, D. J., K. Mills, E. Hestness, L. Cabrera, J. Plane and J. R. McGinnis. 2020. Teacher Change Following a Professional Development Experience in Integrating Computational Thinking into Elementary Science. *Journal of Science Education and Technology*. Vol. 29. pp. 174–188.

Lee, I., S. Grover, F. Martin, S. Pillai and J. Malyn-Smith. 2020. Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering and Mathematics Education. *Journal of Science Education and Technology.* Vol. 29. pp. 1–8.

NEP. 2020. *National Education Policy* 2020. Ministry of Education, Government of India. Retrieved on 10 June 2023 from https://www.education.gov.in/sites/upload_files/mhrd/files/NEP_Final_English_0.pdf

Qualls, J. A. and L. B. Sherrell. 2010. Why Computational Thinking Should be Integrated into the Curriculum. *Journal of Computing Sciences in Colleges*. Vol. 25, No. 5. pp. 66–71.

Reichert, J. T., D. A. C. Barone and M. Kist. 2020. Computational Thinking in K-12: An Analysis with Mathematics Teachers. *Eurasia Journal of Mathematics, Science and Technology Education.* Vol. 16, No. 6.

Repenning, A., D. Webb and A. Ioannidou. 2010. Scalable Game Design and the Development of a Checklist for Getting Computational Thinking into Public Schools. Paper presented at The 41st ACM Technical Symposium on Computer Science Education, SIGCSE, Milwaukee, USA.

Riley, D. D. and K. A. Hunt. 2014. *Computational Thinking for the Modern Problem Solver.* CRC Press.

SENGUPTA, P., A. C. DICKES, A. V. FARRIS, A. KARAN, D. MARTIN AND M. WRIGHT. 2015. Programming in K-12 Science Classrooms. *Communications of the ACM*. Vol 58, No. 11. pp. 33–35.

SENGUPTA, P., J. S. KINNEBREW, S. BASU, G. BISWAS AND D. CLARK. 2013. Integrating Computational Thinking with K-12 Science Education Using Agent-Based Computation: A Theoretical Framework. *Education and Information Technologies.* Vol. 18. pp. 351–380.

SHUTE, V. J., C. SUN AND J. ASBELL-CLARKE. 2017. Demystifying Computational Thinking. *Educational Research Review*. Vol. 22. pp. 142–158.

TABESH, Y. 2017. Computational Thinking: A 21st Century Skill. *Olympiads in Informatics*. Vol. 11, No. 2. pp. 65–70.

TUCKER-RAYMOND, E., M. CASSIDY AND G. PUTTICK. 2021. Science Teachers Can Teach Computational Thinking Through Distributed Expertise. *Computers and Education*. Vol. 173. 104284.

WATERMAN, K., L. GOLDSMITH, M. PASQUALE, E.P. GOLDENBERG, J. MALYN-SMITH, A. DEMALLIE AND I. A. LEE. 2018. Integrating Computational Thinking into Elementary Mathematics and Science Curriculum Materials and Instruction. In Pixel (Ed.), *Conference Proceedings: The Future of Education 2018.* Florence, Italy: Libreria Universitaria Edizioni.

WEINTROP, D., E. BEHESHTI, M. HORN, K. ORTON, K. JONA, L. TROUILLE AND U. WILENSKY. 2016. Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*. Vol. 25. pp. 127–147.

WING, J. M. 2006. Computational Thinking. *Communications of the ACM*. Vol. 49, No. 3. pp. 33–35.

WING, J. M. 2014. Computational Thinking Benefits Society. *Social Issues in Computing.* Available at: http://socialissues.cs.toronto.edu/2014/01/computational- thinking/

YADAV, A., H. HONG AND C. STEPHENSON. 2016. Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends*. Vol. 60. pp. 565–568.

YE, H., B. LIANG, O. L. NG AND C. S. CHAI. 2023. Integration of Computational Thinking in K-12 Mathematics Education: A Systematic Review on CT-based Mathematics Instruction and Student Learning. *International Journal of STEM Education.* Vol. 10, No. 1.