A Review of the Importance of Computational Thinking in K-12

Katharine McClelland and Lori Grata Instructional
Technology and Leadership doctoral student Duquesne
University
Pittsburgh, PA USA
mcclellandk@duq.edu
gratal@duq.edu

***Abstract*—In the K-12 setting, it is important for students and teachers to recognize that computational thinking is more than just using technology or computer science. It is a mindset, a way of approaching difficult problems. Approaching all content areas through this lens will help ensure all learners are exposed to these valuable skills and are able to be successful in our ever-evolving global society. This paper reviews computational thinking in a K-12 setting, considering all content areas and inclusive practices.**

Keywords-computational thinking, K-12.

I. INTRODUCTION

Computational thinking opens doors for more than just students of technology. It is a way of thinking through problems and processing the steps which can lead to a solution, helping develop the capacity and limits of computing [12]. In the K-12 setting, it is important for students and teachers to recognize that computational thinking is more than just using technology or computer science. It is a mindset, a way of approaching difficult problems. Yadav, Hong, and Stephenson state that "the essence of computational thinking involves breaking down complex problems into more familiar/manageable sub-problems (problem decomposition), using a sequence of steps (algorithms) to solve problems, reviewing how the solution transfers to similar problems (abstraction), and finally determining if a computer can help us more efficiently solve those problems (automation)" [13].

With the constant evolution of technology, it is imperative that teachers empower their students to become digital citizens and encourage them to take ownership of their learning. Students may be born with technology in hand, but they must be guided so that they know how to use it appropriately. Yadav, Hong and Stephenson stress that computer science plays a large role in our current society and helps to keep it connected [13]. Therefore, by introducing computing ideas like computational thinking to students early, we can help children become more than just consumers of technology; they can use the tools to someday make an impact on the world.

Computational thinking can be defined as the process of taking a difficult problem and breaking it apart into multiple little problems which we know how to solve. Wing states that computational thinking "involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" [12]. As computational thinking has continued to evolve, the International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) collaborated with members of K-12, higher education, and members of industry to develop an operational definition of computational thinking [5]. From this collaboration, we can define computational thinking as a problem-solving process that includes the characteristics of formulating problems in a way that enables one to use a computer and other tools to help solve them, logically organizing data, analyzing data, representing data through abstractions, and automating solutions through algorithmic thinking. Additionally, computational thinking is identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources. Computational thinking can generalize and transfer the problem-solving process to a wide variety of situations, such as confidence in dealing with complexity, persistence in working with difficult problems, tolerance for ambiguity, the ability to deal with open-ended problems, and the ability to communicate and work with others to achieve a common goal or solution [5].

When applied across different content areas, computational thinking influences how students approach and solve problems. In providing various ways to approach problems, computational thinking helps to ensure success for the problem-solver [11]

Providing students with the tools and supports to find new or unique methods to solve problems will also strengthen students' confidence in their ability to problem-solve. Educators should continually work towards instilling this sense of agency within their students, directly affecting their ability to take control of their lives both in and out of the classroom and into the future [7]. Sneider, Stephenson, Schafer and Flick point out that when students approach a problem with a background of computational thinking, their knowledge can help them see the systems that are before them and develop new problem-solving skills within any content area [11].

In the sections to follow, we will cover the significance of computational thinking in mathematics and science in Section II, Section III will discuss computational thinking in special education, English and history will be discussed in Section IV and electives will be covered in Section V with our conclusion being found in Section VI.

 ISBN: 978-1-61208-619-4

II. MATHEMATICS AND SCIENCE

In K-12 learning, especially within the mathematics and sciences, embedding computational thinking ideas aids students as they work to expand their understanding of concepts and processes. It is important to recognize that this is not a new idea or fad, but rather a foundational concept that has recently earned the attention of educators and researchers. In fact, some of the same skills that are classified as computational thinking are woven throughout the Mathematical Practices (MP) and the Science and Engineering Practices (SEP) found within the Common Core State Standards (CCSS), as shown in Figure 1 [8]. These practices are designed to support students' learning and understanding of mathematical, scientific, and engineering practices throughout their entire education, starting in kindergarten. In fact, most teaching resources, including textbooks and online resources integrate real world applications of these practices and computational thinking skill sets throughout their lessons, activities, and explorations. A few examples include Google for Education, Code.org, and ISTE.

Alignment between mathematical practices and scientific and engineering practices.

Mathematical Practices (MP)	Science and Engineering Practices (SEP)
1. Making sense of problems and persevering in solving them	1. Asking questions and defining problems 3. Planning and carrying out investigations
2. Reason abstractly and quantitatively	2. Developing and using models 3. Planning and carrying out investigations 5. Using mathematics and computational thinking
3. Construct viable arguments and critique the reasoning of others	5. Using mathematics and computational thinking 6. Constructing explanations and designing solutions 7. Engaging in argument from evidence 8. Obtaining, evaluating, and communicating information
4. Model with mathematics	2. Developing and using models 3. Planning and carrying out investigations
5. Use appropriate tools strategically	2. Developing and using models 3. Planning and carrying out investigations 4. Analyzing and interpreting data
6. Attend to precision	3. Planning and carrying out investigations 8. Obtaining, evaluating, and communicating information
7. Look for and make use of structure	4. Analyzing and interpreting data 6. Constructing explanations and designing solutions 7. Engaging in argument from evidence
8. Looking for and expressing regularity in repeated reasoning	5. Using mathematics and computational thinking 6. Constructing explanations and designing solutions

Figure 1. Alignment between mathematical practices and scientific and engineering practices.

Although mathematics and science are assumed to be the most natural areas for computational thinking, it spans across all other curricula as well. Czerkawski explains that this is because computational thinking is merely a problem-solving skill for all disciplines and can be taught through integration in the content area or exclusively teaching the skills [3]. Barr, Harrison, and Conery explained that by integrating computational thinking in the K-12 curriculum across content areas, students are able to learn these important skills in a non-traditional way that enables students to internalize them, thus making it natural for students to connect the knowledge across content and apply the skills in different situations [1]. Figure 2 provides a visual representation which supports Barr, Harrison and Conery's point that every student should be learning computational thinking as it affects core subjects like reading writing and math; it is a way of thinking, processing, and problem-solving [1] [2].

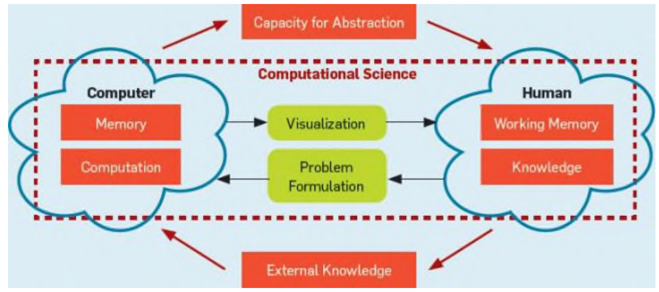

Figure 2. Connecting Computational Thinking to Life

Computation thinking helps build skills that all levels of learner need, including "confidence in dealing with complexity, persistence in working with difficult problems, tolerance of ambiguity, the ability to deal with open-ended problems, and the ability to communicate and work with others to achieve a common goal or solution" [1]. In fact, Deschryver and Yadav take this point one step further as they argue for the need of both "new literacies and computational thinking to promote creative thinking" across disciplines in an attempt to bridge the divide between traditionally creative content areas (music, art, and writing) and scientific areas (math, science, engineering) [4]. By embedding learning activities using collaboratively defined literacies and the incorporation of computational thinking skills, foundational skills can be developed to help scaffold learning and foster creative thinking amongst learners, helping to avoid narrow interpretations and approaches to learning.

III. SPECIAL EDUCATION

Teachers of special education use computational thinking in their day-to-day work; they are trained to see patterns between students and behaviors. Special education teachers use this skill set to teach students to look at complex problems in different ways, applying the content and computational thinking skills as they problem-solve. This is particularly important as Israel, Wherfel, Pearson, Shehab, and Tapia point out that students with disabilities are underrepresented in the fields of science, technology, engineering, and mathematics (STEM) [6]. So, by including computational thinking skills within the K-12 curriculum, students both with and without learning disabilities will grow more through their exposure to these skills and approaches to problem-solving. One integral element of computational thinking is a collaborative learning experience, highlighting students' attributes and showcasing their respective strengths [6].

IV. ENGLISH AND HISTORY

In the areas of English and history, it is natural to collaborate to build students' communication, writing, and reading skills, which align nicely with the concepts of computational thinking, including conceptualization (of the problem), abstraction (of important details), and creativity (when developing the solution/outcome). Shaikh explains that in the English language arts, history, and social studies classrooms computational thinking can be used to teach students how to use a piece of software to create a product [10]. Through this collaboration, the lesson can be extended to include computational thinking skills, allowing the students to be challenged. An example that highlights the integration of computational thinking in a history class is an assignment where the student compares events of ancient times with a his/her current life in a blog [1].

As an alternative example, when discovering problems within literature or history, students could be encouraged to expand a problem and find new ways of looking at it [9]. Computational thinking can aid K-12 educators in their classroom by linking the current educational objectives to classroom practices [1].

V. ELECTIVE CLASSES

Elective classes in a K-12 setting are often overlooked but can be instrumental in teaching computational thinking within a K-12 setting. Computer programing is a common elective that students can take to learn computational thinking when offered, but computational thinking does not need to be limited to just one elective. Like in the core subjects, electives can be useful for engaging students in a new way of thinking. According to the Department of Labor, there are an estimated 1.4 million computer-related jobs that will be available by 2020. Therefore, encouraging students to peruse electives in computing will help build early interest which will help fulfil these estimated jobs (as cited in [6]). Music is another elective area in which computational thinking can be taught and used. One example is having students use scratch.com to create musical instruments to study pitch. Using scratch, the students learning pitch are using computational thinking through abstractions [1].

VI. CONCLUSION

By integrating computational thinking in a variety of disciplines in K-12, students will learn computational thinking throughout the school day, having the opportunity to use and combine their skills within different subject matter [1]. According to Sanford and Naidu, a student can use computational thinking to extend his/her thinking beyond the obvious solution, regardless of the class, as it encourages student initiative and innovation [9]. Through this approach, students become prepared and excited to answer the "what if" questions that are proposed in class, which is necessary when developing students as life-long learners.

REFERENCES

[1] D. Barr, J. Harrison, and L. Conery, "Computational thinking: A digital age skill for everyone," in Learning & Leading With Technology, 2011, vol. 38 issue 6, pp. 20-23.

[2] S. Cooper, L.C. Pérez, and D. Rainey, "Education K-12 computational learning," in Communications of the ACM, 2010, vol. 53, issue 11, p. 27

[3] B.C. Czerkawski, "Classroom implementations of computational thinking: Examples from education majors," in Proceedings of the World Conference on E-Learning, USA, 2016, pp. 151-156.

[4] M.D. Deschryver, and A. Yadav, "Creative and computational thinking in the context of new literacies: Working with teachers to scaffold complex technology-mediated approaches to teaching and learning," in Journal of Technology and Teacher Education, 2015, vol. 23, issue 3, pp. 441-431.

[5] International Society for Technology in Education, and Computer Science Teachers Association, "Operational definition of computational thinking for K–12 education." (2011).

[6] M. Israel, Q.M. Wherfel, J. Pearson, S. Shehab, and T. Tapia, "Empowering K-12 students with disabilities to learn computational thinking and computer programming," in Teaching Exceptional Children, 2015, vol. 48, issue 1, pp. 45-53. doi: 10.117/00400599155594790.

[7] J. Malyn-Smith, B. Coulter, J. Denner, I. Lee, J. Stiles, and L. Werner, "Computational thinking in K-12: Defining the space," in Proceedings of the Society for Information Technology & Teacher Education International Conference, USA, D. Gibson & B. Dodge, Eds., 2010, pp. 3479-3484.

[8] R. Mayes, and T.R. Koballa, "Exploring the science framework: Making connections in math with the common core state standards," in Science Scope, 2012, vol. 36, issue 4, pp. 15-22.

[9] J.F. Sanford, and J.T. Naidu, "Computational thinking concepts for grade school," in Contemporary Issues in Education Research (Online), 2016, vol. 9, issue 1, p. 23.

[10] S. Shaikh, "Should we be teaching computational thinking in schools?," in Fractus Learning, 2016.

[11] C. Sneider, C. Stephenson, B. Schafer, and L. Flick, "Computational thinking in high school science classrooms," in The Science Teacher, 2014, vol. 81, issue 5, pp. 53-59.

[12] J.M. Wing, "Computational thinking," in Communications of the AMC, 2016, vol. 49, issue 3, pp. 33-35.

[13] A. Yadav, H. Hong, and C. Stephenson, "Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms," in Tech Trends, 2016, vol. 60, pp. 565-568, doi: 10.1007/s11528-016-0087-7.

 ISBN: 978-1-61208-619-4