Determining Pedagogically Sound Methods of Teaching and Learning Computational Skills

Cheryl Farren Tkacs
Instructional Designer
The Pennsylvania State University
University Park, PA, USA
Email: cft10@psu.edu

***Abstract*— If we assume even the youngest person can be guided and helped to become a computational thinker, is placing a computer or tablet in their hands the best way they will learn? This paper explores the literature to determine what computational skills learners need to function in the 21st century. Relying on sound pedagogical practices, it will also look at the types of technology and other non-computer devices that are currently being used and whether they are the most appropriate for the age level in developing computational thinking. Based on current research, a list of advantages and disadvantages of children using computers will be explored with recommendations for best practices offered for safe use.**

***Keywords-computational thinking*; *pedagogy*.**

I. What is Computational Thinkning?

The term computational thinking (CT) has become a popular term in recent years when Jeannette Wing used it to describe a fundamental skill that will be needed by everyone in the 21st century. Simply put at the time, she referred to it as "thinking like a computer scientist" [1]. She later refined the definition with input from colleagues to state: "Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer- human or machine- can effectively carry it out [2]. Computational thinking describes a cognitive approach that encompasses a list of abilities used in a problem solving process. For education, it allows the learner to move beyond tasks involving the lower order thinking skills and concentrate on developing more critical thinking and problem solving expertise that is based on concepts of computer science. The evolving definition of CT includes four cornerstones integral to the thought process. They include *decomposition* which is a breaking down of complex problems into smaller components, *pattern recognition* that looks for similarities within problems, *abstraction* that requires focus on the important information, and *algorithms* to develop a step by step solution to the problem [3]. These four skills are intertwined and separating them in the process would cause faulty outcomes in both programming for computer science and the thought processes in solving problems in other content areas.

In 2011, The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) in partnership with leaders in education and industry, developed an operational definition of computational thinking [4]. It expands on the simpler definition first advanced by Wing and her colleagues. The operational definition states: Computational thinking (CT) is a problem-solving process that includes (but is not limited to) the following characteristics:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them
- Logical organizing and analyzing data
- Representing data through abstractions such as models and simulations
- Automating solutions through algorithmic thinking
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources
- Generalizing and transferring the problem solving process to wide variety of problems

This operational definition includes the statement that the above skills are enhanced by attitudes that include:

- Confidence in dealing with complexity
- Persistence in working with difficult problems
- Tolerance for ambiguity
- The ability to deal with open-ended problems
- The ability to communicate and work with others to achieve a common goal or solution

Although its origins are in computer science, computational thinking has expanded and has application in other disciplines as well. Students use this logical approach to solving problems in many subject areas already. Whether it is math, language, the arts, or technology, there are complex problems to break down, steps created to accomplish the task, key components to be focused on, and research on how other projects with similar elements have been solved [5]. As Wing noted in her blog in 2016, great strides have been made in the 10 years since she first brought computational thinking to the forefront of computer science. We still need to look at how best we should be

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-619-4

using computer technology in the classroom. More specifically, determining what is the most effective way to teach and learn computational thinking to all age and levels of learners [1].

Computational thinking is a process that allows us to tackle the solution of complex problems by implementing a process that breaks down the problem into a series of steps to develop solutions in a way that both humans and computers can understand. The primary focus is not programming but conceptualizing a method [6]. It is a combination of problem solving and critical thinking to create new ideas and solutions while using the higher level thinking skills. There are four main components to this process: decomposition, abstraction, analyzation, and algorithms.

Decomposition is the first step in the procedure and, as the term suggests, it is the process of deconstructing a complex problem into small parts. Being able to identify the important details is the first step in thinking abstractly thereby allowing learners to construct a solution that may be out of their normal area of expertise [7]. The ability to think critically is transferable to many disciplines both in K-12 and in higher education. Most problems are not isolated and being able to analyze relationships between problems is a matter of formulating a system of codes.

Thinking computationally requires the ability to think abstractly. Can the problem be explained or represented using a model or a simulation? Abstraction can be used to define patterns, make generalizations, or find properties that are common among the elements of a problem. It is ultimately the ability to transfer the scale and complexity to larger problems [8]. Abstraction hides the details. Direction must be on filtering only the key elements and being able to ignore extraneous details. Mastering the ability to sift through layers of information and get to the heart of the problem is a skill that is essential in all logical thinking processes leading to a confidence in dealing with more complex problems [4].

Computational thinking helps in analyzing possible solutions in the most effective method. Being able to review resources allotted and effectively use those resources can produce a cost-effective solution to the problem at hand often saving time and money in the process. Because computational thinking can be more tool oriented that other types of thinking, the combination of human thinking skills and computer technology can be a formidable solutions team [9].

Whether we realize it or not, we constantly use algorithms in making decisions or solving problems. An algorithm is nothing more than a series of steps to follow in completing a task or produce a solution. When using algorithms, the chances of making a mistake are minimalized while the chances for accuracy and success are maximized. Computational thinking is an extension of algorithmic thinking as it builds upon and incorporates many levels of abstractions in seeking solutions to problems. As such, it is an integral part of all school curriculums and, arguably, part of our everyday lives. Educators and employers assume that a learner has acquired some generic and personal skills through the process of education. These include areas of communication, problem solving, quantification, analytics, and synthesizing skills. An improvement or refinement of these skills enhances the academic work of the students and their employability [10] .

This paper explores the literature to determine what computational skills learners need to function in the 21st century. Section II covers a brief history of computational thinking and the first attempts to integrate into a school curriculum. Relying on sound pedagogical practices, Section III will look at the types of technology and other non-computer devices that are currently being used ascertaining whether they are the most appropriate for the age level in developing computational thinking. Studies detailing how children learn is the focus of Section IV. It offers suggestions and possible devices designed with the child’s age level in mind to deliver age appropriate instruction on computational thinking. Finally, a list of advantages and disadvantages for children using computers will be explored with recommendations for best practices and safe use.

II. History of Computational Thinking

Computational thinking has had an influence in such areas as medicine, economics, law, and the humanities. It can be used to recommend online purchases, detect spam email in your Inbox and even personalize the coupons you receive at the local grocery store [11]. It is important to look at events that led to computational thinking being implemented across both educational curriculums and now becoming pervasive in our everyday thinking processes.

Computational thinking is not a new phenomenon. As early as the 1950s, computer experts were advocating the value of coding for deconstructing the components and using computer analysis to solve problems. Alan Perlis, along with his colleagues at Carnegie Institute of Technology (Carnegie Mellon University) coined the term *algorithmizing* to describe how humans do things arguing that it should become an integral part of our culture [12]. Eric Dijkstra, another forerunner in the field of computing, believed that the distinct nature of computing comes from its unique way of algorithmic thinking that could use natural language to connect problems and solutions [13]. The idea of algorithmic thinking becoming a multi-purpose tool was also being argued by many in the field as leading to higher order cognitive skills useful in multiple disciplines [14].

During the same era, the idea of computational thinking was gaining momentum in education with educators proposing these multi-purpose tools be implemented into the curriculum. Researchers expressed opinions on how computers could make teaching math, languages, music or any subject that require a proficiency of both mechanical and intellectual skills more dynamic using the frameworks provided by the computational thinking in computer science [15]. Seymour Papert was a great advocate of procedural thinking in the construction of knowledge and was one of the first to use the term computational thinking [16]. Papert's work was influential in the field of computers and education focusing his work on how children learn; encouraging student centered and project based discovery learning using technology. As a result of his work in *Mindstorms*, many people argued that learning to program developed cognitive skills that increased the ability to problem solve in many disciplines [16], [17]. His critics said there was no empirical evidence to support these claims of transfer and enhancement of cognitive skills to all domains [18]. The 1980s, with the invention of the supercomputers to aid in calculations and simulations, amplified the need for what was now being referred to as computational thinking in describing the mindset that was developed while using computational science [19]. Through the 1990s, computers began finding their way into schools and homes under the auspices of access to simulations, basic programming, preparation to enter Science, Technology, Engineering, and Math (STEM) fields and preparing students with 21st century skills [14].

Jeannette Wing entered the conversation when she reintroduced the term computational thinking back into academia. She followed the thinking of previous arguments that computational thinking was a tool that has its foundation in information processes and the ability divide a problem into its parts, concentrate on the important components, and develop a strategy for solving the problem thinking [1]. Wing's promotion of computational thinking was joined by such organizations such as Computer Science Teachers Association (CSTA), Computing at School (CAS) and Australian Curriculum Assessment and Reporting Authority (ANCARA) who outlined their own vision for computational thinking in the schools. They all included aspects of skills, attitudes, techniques, and lesson plans for developing logical reasoning, algorithmic thinking, decomposition, abstractions, and evaluation that schools are still trying to implement 10 years later [14].

Not all agree that computational thinking is the best method of problems solving including Papert who expresses his receptiveness to other approaches [16]. Other types of thinking have been considered indispensable in science and technology and have also been supported by educators including engineering thinking, logical thinking, systems thinking, rational thinking, and ethical thinking to name just a few [14]. Nevertheless, computational thinking is establishing itself in the K-12 environments where students are learning programming by exploring and practicing; building on Papert's vision of constructivism.

Making computational thinking an integral part of any curriculum still has many obstacles to overcome. Discussions and research on pedagogical issues, assessment of CT learning objectives, and the most important issue of deciding what should be taught and when it should be taught is just part of the work that must be done to support the belief that CT has a place in all disciplines not just computer science.

III. CHILDREN AND COMPUTERS

There is no doubt that computers have transformed the lives of people all over the world. In order to compete in this world, it has become necessary to have a level of competence using technology and this particularly includes our children as they prepare for a world not even invented as yet. Computers are shaping children's lives both at home and in school. They play games, use social media to connect with friends and family, attend class online, do homework, access the Internet, and watch videos. As early as 1994, Congress decided that technology can aid students in meeting a higher standard of learning and enacted the *Goals 200 Educate America Act* and the *Improving America's Schools Act*. As a result, the percentage of schools with computer access to the Internet increased from 35% to 95% (U.S. Department of Education, 2000). In the *Current Population Survey* conducted in 2001, several key findings were noted about computer usage among children ages 5 to 17 [19].

- 90% of children and adolescents use computers. (47 million persons).
- Computer usage starts early. 75% of five years olds use computers and about 25% of them use the Internet.
- Computer and Internet usage is divided along demographic and socioeconomic lines with children of more highly educated parents having the higher use rate.
- There was no difference in sexes in overall computer usage as compared to earlier statistics where boys had a higher rate of usage.
- More children and adolescents use computers at school (81%) than at home (65%).

Comparing those statistics to the 2012 U.S. Census Bureau, the age span has changed to include 3 to 17 year olds with a 70% access to the Internet from inside and outside the home. In 2012, with nearly every school having computers and two-thirds of children having Internet and computer access in their homes, it is apparent that computers had become an integral part of children's lives in a short span of time. The number of children having access to computers and the Internet is growing exponentially every year. It is also important to note that these are statistics for only the United States [20].

If we look beyond the U.S. borders, approximately one-third of the world's population is under the age of 18. In third world countries they make up half of the population while they are less than 25% in industrialized nations. Around the world, children face many challenges from basic survival to discrimination and exploitation. In recent years, due to the greater availability of media through satellites,

Copyright (c) IARIA, 2018. ISBN: 978-1-61208-619-4

more and more young people in third world countries have access to computers, TV, and the Internet and are exposed to information from around the world. It is that access to technology that can give them the chance at education and being able to make a difference in their world [21].

With computers engrained in children's lives, it is important to understand how computers can both enrich and benefit students as well as how it hinders their growth and development.

A. Disadvantages of Computer Use by Children and Adolescents

Opponents of young children using computers feel that parents, educators, and psychologists should take a more in-depth look at the risks of supplying computers at too early of an age. Their criticism includes children having access to violent games, inappropriate content, and aggressive advertising which can adversely affect their relationships with other children and the adults in their lives. They cite a 1998 National Science Board report that overuse of computers by children can create individuals that will not be able to cope with reality and the demands of personal commitments [22].

Research has shown that there are a number of physical and emotional concerns that can arise from prolonged use of computers by children and adolescents. For toddlers, too many "bells and whistles", bold colors, and flashing lights can overstimulate and the child becomes irritable and cranky [15]. Smaller children need to have more human interaction in order to learn social skills to communicate effectively (verbally and non-verbally). Without these social skills, children find it difficult to read subtle signals through reading body language and personal appearance [23].

An important side effect of prolonged computer use is that there is too much sedentary time involved. Children's posture and bone growth can be stunted and the lack of physical play can result in week muscles and obesity [24] [11]. Results can range from injuries to backs, wrists, and legs to seizures in children suffering from photosensitive epilepsy. Children with computers in their rooms get two to five hours less of sleep that their parents did at the same age [25]–[27]. Behavioral problems, including aggressive behavior, have been reported in children that engage in online game playing for long intervals [28], [39]. While older children can improve their visual spatial skills when using or playing computer games, too much on-screen time can negatively impact a toddler's eyesight. The nerves and eye muscles are not sufficiently developed and computer vision syndrome can result [29].

The disadvantages of computer use among children and adolescents is not insurmountable. Practical suggestions, common sense usage, and monitoring by parents can ensure that children can safely and effectively use computers to increase their intellectual and physical development so they can succeed in whatever world they find themselves in the future.

B. Recommendations for Children Using Computers

In order to provide the safe and intellectually engaging experiences for children using the computer at home and in educational situations, The Future of Children organization offered some guidelines to protect children's physical and mental development while still allowing them the freedom to explore, communicate, and learn with technology. Their recommendations were published in the journal *Children and Computer Technology* and the main suggestions include:

- More public and private research to assess the effects of extended computer use on children's physical, intellectual, social and psychological development.
- Parents, teachers, and other adults that work with children should limit time spent with computers and supervise the content they are accessing.
- Dialog among researchers, software developers, and government agencies should be encouraged and supported to create high quality content for children.
- Education agencies should research, refine, and adopt age appropriate guidelines for children's computer fluency.
- Teachers both in education technology programs as well as classroom teachers should be provided with professional development workshops that are focused on the training and skills they need to use age appropriate technology in the classrooms.

In addition to the above list, *The Future of Children* organization also made recommendations to help narrow the disparity of computer and Internet access between socio-economic groups and they addressed the need for universal design to be included so children with special needs will have the same advantages in using computers to learn, discover, participate, and compete in the world.

IV. TEACHING COMPUTATIONAL SKILLS

If we accept the premise that children should be taught computational thinking, the next step would be to decide the most effective way to teach and learn CT. Parents and educators have a responsibility to use the research to determine what concepts students can best learn and when in their developmental stage. What should we teach and when? Toddlers are handed iPads and they explore by pressing icons to see what will happen. As the child enters teen years, the dependence on electronic devices is evident as you watch them hunched over staring at the screen, texting rather than interacting face to face with their friends rarely being farther than a hand's reach away from their phones. This is not necessarily learning computational thinking. In addition, research conducted by health care professionals remind us of the benefits and hazards that can result from overuse and misuse of technology in both children and adolescents.

With the knowledge gained from studies on how children learn and focusing on using devices, software, and STEM toys that are age appropriate, this next section will look at options for teaching and reinforcing the understanding of computational thinking concepts for children and adolescents. The idea of coding is the element that most of these devices and software have in common. They teach children to think logically by applying the method of analyzation, decomposition, application of solutions and then generalizing those solutions to new situations are key objectives in these lessons.

A. Toddlers

Today's parents want to give their child an intellectual head start by placing electronic devices in the child's hands to strengthen their computer and problem solving skills. And yet, research supports the fact that children under the age of two do not have the hand-eye coordination to hold a device, move a mouse, tap an icon, follow the action on the screen or have the attention span to understand what is happening. Experts believe that toddlers are more in need a more hands-on relationship with the world and people around them.

This does not mean children as young as three years old can't learn to think critically and or will fall behind their peers if not given access to computer devices. Children at this age are learning creativity and developing their motor skills. There are many STEM toys available on the market that foster computational thinking

through tactile play. A few of these STEM toys that fulfill both the intellectual and physical needs of 3 to 5-year-old children promoting discovery and problem solving skills that are key in computational thinking include manipulative robots that can be programmed in a variety of ways.

Think & Learn Code-a-Pillar teaches the basics of coding. The segments of the *Code-a-Pillar* contain chips that area embedded with the commands turn right, turn left, make a sound, and more. Children can separate and reconnect them in any order and the toy will carry out the sequence. *Cubetto* is a small square smiling robot that, like the *Code-a-Pillar*, will follow a series of commands. Colored blocks represent the commands that *Cubetto* will follow as they are placed in the sequence that the child want the robot to follow. The *Kibo* robot is made of blocks fitted together in a variety of configurations. Once it is built, the robot's body is scanned and pushing a start button carries out the program.

Ozobots are another robot version that area only 1-inch-tall which making them more suitable for age 5 and up. These small robots can move on different types of surfaces including a tablet screen. Paths are programmed using color and the robot follows the colored lines drawn on paper or a screen. Colors correspond to different commands. As the child becomes more advanced, pro-set blocks of code from *Google Blockly*, a library that adds a code editor to web and Android apps using interlocking, graphical blocks , can be used to program the *Ozobots'* movements. *Dash and Dot* are a team of mobile robots that children can use a suite of apps to control. *Dash* is the mobile robot of the duo while *Dot* is stationary. Apps vary in the range of complexity and are run by *Google Blockly*. Puzzles and challenges can be solved by programming *Dash and Dot*.

Engaging in construction-based robotics, even toddlers are learning a wide range of concepts and demonstrate the mastering of various learning outcomes involving computational thinking, robotics, problem-solving, and programming [30]. This list contains just some toys that can provide the opportunity to move through the cornerstones of CT while providing a tactile approach that is not taxing on growing bodies.

B. K-12

As children enter school, they have more options and opportunities to learn and engage in computational thinking both in and out of the classroom. At this stage, learning tools such as toys, puzzles, and games continue to be active means of employing computational thinking. Some board games new to the market including *Robot Turtle Game* and *Code Monkey Island* are designed to teach logic and development and programming skills by using conditional statements, looping, and other operators to move players around the board. *Puzzlet* is another board game that links the student's programming to the way characters move in an app-based world on a tablet. Bringing girls into the world of coding is the focus of items such as *Jewelbots*, the latest incarnation of friendship bracelets. They can be programmed through if/then statements to light up when a friend is near, vibrate if they get a text or a "like" on Instagram, and any number of other programs they can write themselves.

Following progressive steps that are found in board games, such as the ones listed, the student is using increasingly more difficult algorithms in these thinking activities. This type of instructional strategy capitalizes on children's interest and skills [31]. Teachers can add these types of activities to their lessons to connect abstract thinking patterns to real-life situations.

When teaching computational thinking through coding programs, one of the guiding principles is the "low floor, high ceiling" concept. Simply stated, the programming environment being used should be easy enough for a beginners to have success in creating a working program but powerful and complex enough to keep a more advance user engaged [32]. Some of the more popular graphical programming environments include: *Scratch*, *Alice*, *Game Maker*, *Kodu*, and *Greenfoot*. These examples use three stage progression, use-modify-create, to help the learner progress from novice to expert. Older students can use programs like *Snap*, robotic kits, *Arduino*, and *Gogo Boards* as a jumping off point to learning high level programming languages such as Java (a general-purpose computer programming language based on C++), Python (a purpose programming language that emphasizes code readability and a syntax which allows programmers to express directions in fewer lines of code), or Scheme a (programming language that follows a minimalist design philosophy).

For teachers looking to include non-computer lessons to teach computational thinking, there are a host of sites that have pedagogically based lessons to incorporate into almost any discipline for the K-12 environment. A good first stop is Code.org, a non-profit organization whose vision is to have every student have the opportunity to study computer science and especially advocates for women and minorities. They promote computer science and learning computational skills each year through the *Hour of Code* campaign. They provide curriculum guidance with lessons available for elementary, middle, and high school on their website.

Other resources available to teachers looking to include computational and critical thinking skills in their classrooms include Global Digital Citizen Foundation, Barefoot Computing, Computer Science Teachers Association, Exploring Computational Thinking (Google for Education), and CS Unplugged. These sites contain lesson plans and links to additional resources that help teachers incorporate the elements of coding and computational thinking into their courses.

As children grow and develop, parents and educators still need to monitor device use to prevent vision and other stresses that can occur from overuse. It is also important to keep in mind that the computational tools and games that are currently on the market vary in their effectiveness of teaching and addressing engagement with all the components of computational thinking. If we are employing these tools in a K-12 environment, developers need to create additional components to present programs or create new ones that guide the learner through all the competencies of computational thinking and be guided by research strategies on how children learn to problem solve [33].

C. Higher Education

This paper did not look at research that involved computational thinking and computer use in higher education, however, in a few short years students that have been exposed to computational thinking as they progressed through grade school, junior high, and high school will already have those higher order thinking skills and will be expecting universities to continue fostering deeper learning approaches in courses where students will be expected to think critically, conduct problem solving research, collaborate face-to-face or online with their classmates, and participate in more self-paced and directed learning in the courses they take. Some of these students have already arrived on the university's doorstep. Faculty in higher education should be using technology tools in creating course materials and assignments that have real-life application. Some universities are leading the way in using Problem Based Learning (a student centered approach with the teacher facilitating problem solving scenarios), design thinking (teaching students that the best solutions are those that are empathy-driven and end-user-centric), and gaming (where the interface is designed to learn about subject content in order to promote the algorithmic method to solve problems encouraging higher-order thinking). *Makerspaces* are another way of providing the experiences for people of all ages to experiment, iterate, and create in an area that is equipped with technology and other types of tools they can use. The time is now to prepare to meet the needs of the next generation of learners in higher education.

V. CONCLUSION

Computers are tools and how they help or hinder children is dependent on our guidance. In the classroom, it is the responsibility of educators to explore and develop new structural approaches of using technology in ways that support the curriculum goals and learning objectives of the various disciplines. Computers are one way to enhance the traditional curriculum and engage students providing them with a systematic procedure that utilizes computational thinking to solve complex problems. The design practices that are involved with computational thinking does not solely apply to using computers or software programs. These processes, specifically ones involving experimenting, testing and debugging, reusing and remixing, and abstracting can also be applied to the STEM toys discussed in this paper. Once mastered, students will be able to apply the method in all aspects of their lives.

Before children can be introduced to computational thinking, their teachers need to learn how CT fits into core curriculum courses and expand their understanding of how it can be applied. Changes and expansion of programs to include the elements of computational thinking requires vision, planning, and cooperation among administration, teachers, and parents. It also requires a pedagogical knowledge of the mental learning processes that children need to succeed without overtaxing their physical, mental, and emotional well-being. The question of assessment should be addressed as well. If students are part of a curriculum designed to include computational thinking, decisions must be made by educators and administrators detailing what students should be able to do or know and how they will be assessed upon mastering computational skills.

ISTE, CSTA, and the National Science Foundation (NSF) have proposed a Model for Systematic Change for K-12 educators and administrators to use in implementing computational thinking into the curriculums. The strategies' guide includes plans that range from short term (Year 1), mid-term (Years 2 – 5), and long-term (years 6 – 11). These strategies map activities for stakeholders to follow, suggestions for partnerships with national groups, and goals that will help direct and become the agents of change in our schools.

Computational thinking has gone far beyond teaching computer science since its inception in the 20th century. While much of what we are teaching in our schools today will be obsolete in 5 to 10 years, the ability to think critically and creatively are the skills that are, and will be, valued as students move from academia and into the workforce.

REFERENCES

[1] J. M. Wing, “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3. pp. 33–35, 2006.

[2] J. Cuny, L. Snyder, and J. M. Wing, “Demystifying computational thinking for non-computer scientists,” *Unpubl. Manuscr. progress, 2010. [Online]. Available: http//www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf.* [Accessed on Sept. 12, 2017].

[3] C. C. Selby and J. Wollard, “Computational thinking : The developing definition,” *ITiCSE Conf. 2013*, pp. 5–8, 2013.

[4] ISTE and CSTA, “Operational definition of computational thinking,” *Report*, p. 1030054, 2011.

[5] Barefoot Computing, “Computational thinking: How do we think about problems so that computers can help?,” 2014. [Online]. Available: http://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/. [Accessed on Sept. 28, 2017].

[6] A. Yadav, “Computational thinking and 21st century problem solving,” 2011.

[7] H. S. Sarjoughian and B. P. Zeigler, “Abstraction mechanisms in discrete-event inductive modeling,” in *Winter Simulation Conference Proceedings*, 1996, pp. 748–755.

[8] J. Uhlman and A. Aho, “Foundations of computer science in C,” *Found. Comput. Sci.*, pp. 776–786, 1994.

[9] D. Barr, J. Harrison, and L. Conery, “Computational thinking: A digital age skill for everyone,” *Learn. Lead. with Technol.*, vol. 38, no. 6, pp. 20–23, 2011.

[10] D. Bradshaw, “Transferable intellectual and personal skills,” *Oxford Rev. Educ.*, vol. 11, no. 2, pp. 201–216, 1985.

[11] J. M. Wing, “Computational thinking benefits society,” *J. Comput. Sci. Coll.*, vol. 24, no. 6, pp. 6–7, 2014.

[12] A. Newell, A. J. Perlis, and H. A. Simon, “Computer science,” *Science (80-.).*, vol. 157, no. 3795, pp. 1373–1374, 1967.

[13] E. W. Dijkstra, “Programming as a discipline of mathematical nature,” *Am. Math. Mon.*, vol. 81, no. 6, p. 608, Jun. 1974.

[14] M. Tedre and P. J. Denning, “The long quest for computational thinking,” *Koli Call. Conf. Comput. Educ. Res.*, pp. 120–129, 2016.

[15] W. Feurzeig, “Computer systems for teaching complex concepts,” Cambridge, MA, 1969.

[16] S. Papert, *Mindstorms: Children, computers and powerful ideas*, vol. 1. 1983.

[17] R. E. Mayer, J. L. Dyck, and W. Vilberg, “Learning to program and learning to think: What’s the connection?,” *Commun. ACM*, vol. 29, no. 7, pp. 605–610, 1986.

[18] M. Guzdial, “Education paving the way for computational thinking,” *Commun. ACM*, vol. 51, no. 8, p. 25, 2008.

[19] M. Debell and C. Chapman, “Computer and internet use by children and adolescents in 2001: Statistical analysis report,” Jessup, MD, 2004.

[20] M. K. Shields and R. E. Behrman, “Children and computer technology: Analysis and recommendations,” vol. 10, no. 2, pp. 4–30, 2000.

[21] S. Gigli, “Children, youth and media around the world: An overview of trends & issues,” Washington, D.C., 2004.

[22] “Computers and children,” *Alliance Child.*, pp. 1–8, 2000.

[23] K. Subrahmanyam, P. Greenfield, R. Kraut, and E. Gross, “The impact of computer use on children’s and adolescents’ development,” *J. Appl. Dev. Psychol.*, vol. 22, no. 1, pp. 7–30, Jan. 2001.

[24] E. de Jong, T. L. S. Visscher, R. a HiraSing, M. W. Heymans, J. C. Seidell, and C. M. Renders, “Association between TV viewing, computer use and overweight, determinants and competing activities of screen time in 4 to 13-year-old children,” *Int. J. Obes. (Lond).*, vol. 37, no. 1, pp. 47–53, 2013.

[25] L. Straker, B. Maslen, R. Burgess-Limerick, P. Johnson, and J. Dennerlein, “Evidence-based guidelines for the wise use of computers by children: Physical development guidelines,” *Ergonomics*, vol. 53, no. 4, pp. 458–477, 2010.

[26] P. C. Kendall, M. S. Khanna, A. Edson, C. Cummings, and M. S. Harris, “Computers and psychosocial treatment for child anxiety: Recent advances and ongoing efforts,” *Depression and Anxiety*, vol. 28, no. 1. pp. 58–66, 2011.

[27] L. Straker, C. Pollock, and B. Maslen, “Principles for the wise use of computers by children,” *Ergonomics*, vol. 52, no. 11, pp. 1386–1401, 2009.

[28] R. B. Williams and C. A. Clippinger, “Aggression, competition and computer games: Computer and human opponents,” *Comput. Human Behav.*, vol. 18, no. 5, pp. 495–506, 2002.

[29] N. Kozeis, “Impact of computer use on children’s vision,” *Hippokratia*, vol. 13, no. 4, pp. 230–231, 2009.

[30] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan, “Computational thinking and tinkering: Exploration of an early childhood robotics curriculum,” *Comput. Educ.*, vol. 72, pp. 145–157, 2014.

[31] S. Y. Lye and J. H. L. Koh, “Review on teaching and learning of computational thinking through programming: What is next for K-12?,” *Computers in Human Behavior*, vol. 41. pp. 51–61, 2014.

[32] S. Grover and R. D. Pea, “Computational thinking in K-12: A review of the state of the field,” *Educ. Res.*, vol. 42, no. 1, pp. 38–43, 2013.

[33] B. Simon, T. Y. Chen, G. Lewandowski, R. McCartney, and K. Sanders, “Commonsense computing: what students know before we teach,” *Proc. Second Int. Work. Comput. Educ. Res.*, pp. 29–40, 2006.