J. Korean Earth Sci. Soc., v. 40, no. 4, p. 340−352, August 2019
https://doi.org/10.5467/JKESS.2019.40.4.340

ISSN 1225-6692 (printed edition)
ISSN 2287-4518 (electronic edition)

Review

Bringing Computational Thinking into Science Education

Young-Shin Park* and **James Green**

Science Culture Education Center, Chosun University, Gwangju 61542, Korea

Abstract: The purpose of science education is scientific literacy, which is extended in its meaning in the 21st century. Students must be equipped with the skills necessary to solve problems from the community beyond obtaining the knowledge from curiosity, which is called 'computational thinking'. In this paper, the authors tried to define computational thinking in science education from the view of scientific literacy in the 21st century; (1) computational thinking is an explicit skill shown in the two steps of abstracting the problems and automating solutions, (2) computational thinking consists of concrete components and practices which are observable and measurable, (3) computational thinking is a catalyst for STEAM (Science, Technology, Engineering, Arts, and Mathematics) education, and (4) computational thinking is a cognitive process to be learned. More implication about the necessity of including computational thinking and its emphasis in implementing in science teaching and learning for the envisioned scientific literacy is added.

Keywords: computational thinking, STEAM, scientific literacy, science education, solve the problem

Computational thinking is not a new term. It has been already known through the work of scientists, but it has been widely cited by Jeannette Wing and an extensive discussion has continued (Nardelli, 2019) through a lot of research as well as talks due to her role at the NSF (National Science Foundation). In this paper, computational thinking practice from the science education view will be discussed as to why it is necessary to include computational thinking in science education, what computational thinking looks like in science teaching and learning, and how CT can be included in science teaching and learning. This paper is to develop a more complete understanding of computational thinking as it applies to the science discipline and the needs of teachers/educators who are expected to prepare students to be problem solvers, envisioned in the 21st century. The statement/claim released in this paper is on the basis of the authors' studies of literature reviews mainly about computational thinking, experience of interacting with science teachers who are interested in computational thinking through professional development program, communication with the researchers who study computational thinking, and their own research about computational thinking. The authors discussed all the pros and cons of computational thinking released from these data resources and made the review direction that is distinct from computer science into science education.

The Purpose of Science Education in the 21st Century

Scientific Literacy with computational thinking

When somebody asks what the purpose of science education is, the answer will be that all people need to be equipped with competencies to be able to make decisions if the issues they face from their daily lives are right or wrong (MOE, 2015; NRC, 2012; Park, 2010). For this, people need to know about what the issues are, they can demonstrate the process of issues logically, if necessary, people argue and develop claims on the basis of evidence from experimentation, but people can give up those products if they are

---

*Corresponding author: parkys@chosun.ac.kr
Tel: +82-62-230-7379

ethically problematic. We call 'scientific literacy' all the abilities necessary to make decisions of whether those issues are right or wrong. But now we need to extend the meaning of scientific literacy. Students need to know how to apply the concepts which they learn. Before, students understand some concepts and they argue why those issues are critical or not with the use of those concepts, but we cannot be confident that students can have abilities to apply concepts to finally produce some solutions practical to the issues/problems they face (NGSS Lead States, 2013; Park and Park, 2018).

To meet this goal of scientific literacy, we teach students science as inquiry (Hannasari, Harahap and Sinulingga, 2017; NRC, 2000; Park, 2010). To equip students with competencies to be literate scientifically through scientific inquiry, students should obtain concepts basically (Contents-ON). In addition, students can demonstrate certain phenomenon by using experimentation through which they collect the data (Hands-ON). Students can discern evidence from data to be supportive of their claims during the process of argumentation (Minds-ON). If Hands-ON is mainly emphasized without Minds-ON, then scientific inquiry can sound like a cookbook system, which is not meaningful for being literate. Therefore, we need to equip students with two different skills; procedural skill for experimentation and thinking skills for argumentation (NRC, 2000; Park, 2010). This thinking is called scientific thinking consisting of logical thinking and critical thinking in order (Kuhn, 1993; Osborne et al., 2010; Park, 2010). For example, in the climate change of STEAM program developed by Park (2012), students had chances to figure out what kind of data they should collect to figure out if there is a change in the climate of their city from the given internet data. Students, therefore, collected the data from the website of the Korean Atmospheric Science Research Center and used some as evidence supporting the claim; there is climate change even in Korea, even the city where they live. Students designed the experimentation to figure out what the greenhouse effect is and what gas is the main factor for global warming. For this, students selected three different black colored materials; black paper, black plastic, and black film. Within the same time lapse, students collected the data of increasing temperature, transforming it, and interpreting it to answer the questions. Students demonstrated the skills of how to collect the data, how to discern evidence from data, and how to use evidence to support their claim. Students had the same claims with different evidence. Students in groups evaluated their claims by supporting, refuting, and redefining, etc. Students experienced logical thinking (provide their claims with evidence) and critical thinking (evaluate other claims and evidence to find out the most ideal ones). Logical thinking comes first and critical thinking comes later, which is called 'scientific thinking' (Kuhn, 1986; 1993; Park, 2006; 2010). These thinking skills take place when students form new concepts/knowledge from their curiosity. Students, now, need to analyze this given situation to find the best solution and they need additional competency (Park and Park, 2018; Wing, 2006; 2008; 2010), which we call computational thinking. Nardelli (2019) made a point that many reserchers agree that the wide popularity of computational thinking after Wing's viewpoint has spoiled the original aim. People stress computational thinking as a new subject, somehow different from computer science, which obscures its meaning. A lot of people seem to argue that computational thinking is new and different. At least, the researchers in this study try to clarify interpretations of 'problem' and 'solution' more broadly with additonal competency of computational thinking. When students face the problem/situation, they can interpret them in different ways.

How can we additionally name computational thinking practices in the science program or science learning as well as teaching? In the climate change section of a STEAM program (Park, 2013), students have chances to design the equipment to consume CO_2 through photosynthesis by using biomass in a bioreactor. After learning the concepts of global warming, the greenhouse effect, and gases with climate change, students come to know that it is very urgent to save the Earth from climate change. Then how can students save the Earth? (decomposing the problem/situation to be more manageable) This is how students come to design the equipment of consuming CO_2 to grow green algae through photosynthesis for the purpose of producing oil as an alternative energy source (abstraction, where students figure out the critical factors of the phenomenon for a solution; green algae through photobioreactor is the abstracted factor). Green algae is considered more efficient than corn as an alternative energy source in Korea. Every chemical factory must construct the photobioreactor which uses biomass to consume CO_2 and produce oil in students' residential area. However, every chemical factory in every residential area of Korea doesn't have to construct a photobioreactor for consuming CO_2. There must be certain conditions of the residential area appropriate to install the photobioreactor to solve the problem, consuming CO_2, which students discuss in evaluating its solution. Students also calculate what shape of photobioreactor is the most efficient for photosynthesis. Students decide to build walls of hexagon shaped mirrors surrounding the photobioreactor for the most efficient photoysynthesis. Then how can we know that photosynthesis is occuring now? Students can know it by its color (this is qualitative way). The color of green algae changes from light green to dark green during two weeks, and this is observable by the naked eye. However, students discuss how and when to deactivate the photobioreactor automatically when the green algae is fully grown and appopriate to produce oil (Students designed the steps/loop indicating when to stop photosynthesis; algorithm and automation). Once the students complete installing the equipment of the photobioreactor, they make a decision of which residential area is appropriate for installing this equipment, how much CO_2 can be consumed as well as how much oil can be produced, and lastly how helpful this solution can be for climate change in the community (simulation). The steps mentioned above from decomposition to simulation are competencies named as some of computational thinking practices (CSTA, 2011; ISTE, 2011; Park and Park, 2018; Weintrop et al., 2016). Other computational thinking practices include data-related practices; data collection, data analysis and data representation which is not introduced but we are familiar with.

The necessity of computational thinking for new era in science education

Students use scientific thinking skills to differentiate evidence from data and develop their claims with the use of selected evidences. This process can be found while students learn concepts from some topics. Students can figure out how those concepts they learn are applicable in their daily lives. Students, however, do not have opportunities to apply those concepts to abstract the phenomenon to be reframed to be a researchable problem, build and evaluate the model for the solution, and run simulation as solution, which could be valuable and useful at the end for the problem. Students can experience 'computational thinking' practices when applying concepts to find out the solution. By doing this, students can be trained to become creative problem solvers for the issues/problems they face. Park and Hwang (2017) named those practical protocols as follows; connecting computing, developing computational artifacts, abstracting, analyzing problems and artifacts, and communicating with collaboration. At this point, communication through collaboration can be differentiated from other typical ones during science classes. Here, those protocols can be observed and measured in the class as the type of computational thinking practices (Park and Hwang, 2017). Park and Park (2018) released what components could make those practices observable quantitatively.

STEAM is the dominating educational policy in Korea for the last 10 years. The government put the emphasis of developing STEAM programs and employing them into the classroom with a lot of funding. But science teachers from K to 12 have been struggling to understand, develop, and implement STEAM programs as envisioned by the government. The MOE (Ministry of Education), however, did not give enough time for science educators to research what STEAM education is and why we need STEAM but it provided a lot of funding for science teachers to develop STEAM programs to be implemented since STEAM has been introduced to science educators as well as science teachers. There have been trials and errors in settling STEAM education down into the context of STEAM programs. Science has been taught through the other 4 disciplines as tools, but surely the other 4 disciplines can be contents in certain points (MOE, 2009). What do we expect from students through STEAM education? Students are expected to design outcomes for the solution, and they learn concepts and apply them to it.

In the past, of science education, we used to equip students with inquiry skills to learn or confirm scientific concepts without the chance of designing engineering practices for the solution (Han and Nam, 2018; Sim et al., 2015). We need to equip students to be more creative problem solvers from the issues they face in their daily lives in the community. Computational thinking is the thought processes related to formulating problems and their solutions so that those solutions are represented in a form which can effectively carried out by an information-process system (Aho, 2011; Nardelli, 2019). The reason why computational thinking is necessary in science education, especially in STEAM education, can be explained as follows; the purpose of science education, 'equipping students to be creative problem solvers rather than answer finders', can be achieved through STEAM programs with the use of computational thinking.

Computational Thinking in Science Education

The computational thinking of computer and technology education must be differently defined from that of science education. In this section, the newly constructed definition of computational thinking in science education will be explored on the basis of the original one in technology education in the papers.

The characteristics of computational thinking in science education

On the basis of our theoretical review about computational thinking (CT after this) in experimental papers, some characteristics of CT in science education can be withdrawn as follows. New definitions about CT in science education can be developed.

CT is an explicit skill in two steps of abstracting the problems and automating the solutions: The defining characteristic of CT is that it vacillates the exploration of problems and their possible solutions. This is why, with society's need to train creative problem solvers, CT can play an important part in STEAM/STEM education. CT is a problem-solving skill that gets progressively more sophisticated as the students get older (CSTA, 2011; ISTE, 2011).

In the seminal article (Wing, 2006; 2008) broke down CT into the 'The two A's of Computational Thinking', abstractions and automation. Abstractions are the mental tools that we use and they are the cognitive and intellectual skills that can be utilized to comprehend problems and then deduce and invent methods of solving the problems. The second A, automation is about the metal tools and they are the physical equipment, like any computer software, which is used to help solve problems. Some examples of metal tools are computers, calculators, thermometers, and graphing software to help visualize the results. Automation is mechanizing our abstractions, abstraction layers, and their relationships (Wing, 2008). The tools enable humans to handle the complexity and permit some of the tasks to be automated. It should be stressed that this is not the same as artificial intelligence, which is an attempt to copy human mental processes, but an amplification of human intelligence. So how can scientific literacy be interpreted with CT? CT can be included into the thought of as an extension of scientific literacy. The extension comes in how the solver tries to find the solution to the problem. According to Wing (2010), computational thinking describes the mental activity in formulating a problem to admit a computational solution, describing a two-step process of formulating the problem and then moving forward to find a solution computationally. Wing (2010) used the words 'problem' and 'solution' with a broad and wide-ranging definition.

These two steps are also illustrated in Hwang (2019), where two experienced elementary teachers came to learn what CT is and how they employ it into their science teaching. Those two participating teachers developed their competencies of revising their curriculum for the purpose of using CT in their science lesson. For example, Mr. Son revised the curriculum of the 'seasonal change' unit, where he extended the latter part of the unit from 2 lessons to 5. Students at grade 5 designed a Korean folk village with different angles of its roof to get the least sunlight to avoid high temperatures in the room during summer. During this process, students calculated which angle of the roof is the best for summer and they tried to figure out how to measure the temperature with the use of Arduino so that they can close the window or curtain when the sunlight is too strong. Students used the program language UNO connected to Arduino for this function. This was a hard task for all of the students at elementary level to complete but at least they had chances to discuss which factors are the critical ones to be considered and how those factors are related each other in finding the solution. Students became to know how to face the problem, how to extract the factors making those problems, and to find the best solution. Students used digital thermometers in the rooms of the folk village and they made Arduino connected to the UNO program to send the signal to close the window or curtain to block the sunlight in summer. Students calculated the angle according to the solar altitude at meridian passage in Korea. At least students discussed what steps they need to consider in making decision to block the Sun's light in summer, which can be 'automation'. Here, Mr. Son provided two steps of CT, one is *abstraction* where students figure out what the problem is and where they learn concepts related to solve the problem; the other is *automation* where students could develop the algorithm to be automated with some points (like set up the digital thermometer connected to UNO and Arduino, calculate the angle of the roof according to the Sun's altitude etc.). Mr. Son indicated that he could form new understandings about CT and developed CT practices by revising the curriculum. In this paper, Mr. Son offered the chances for students to apply science concepts.

Nardelli (2019) stated that CT is not a new subject to teach and what should be taught in school is informatics, but he made a point it is very garbled when CT comes to education. How can we teach CT in the classroom? How can we be sure CT is really effective in education? How can teachers learn to teach it? Nardelli (2019) described CT with two words 'problem' and 'solutions' illustrating CT as follows; Computational thinking is the thought process involved in modeling a *situation* and specifying the ways an *information-processing agent* can effectively operate within it to reach an externally specified (set of) goal(s). The change of 'problem' to 'situation' removes the implication of a required solution. The other difference is the emphasis that the goal(s) must be set externally so that the agent does not delineate the goal(s) itself. At least, CT can be defined into two steps where students can recognize and understand the given situation/problem and specify the ways to operate the information-processing agent operate effectively to reach the goals.

While Wing's 2006 article called for CT to be a part of school education within a range of diciplines it left questions unanswered. How can CT be observed to be taking place? What pedagogical strategy should be implimented to encourage CT in students? To answer these questions there needs to be a clear lexicography. The next section will look at attempts to provide that clarity.

CT consists of concrete components/practices to be observable and measurable: The second edition teacher resources for computational thinking (ISTE and CSTA, 2011) breaks CT down into nine different skills for students to master. Those skills are data collection, data analysis, data representation, problem decomposition, abstraction, algorithms and procedures, automation, simulation, and parallelization. The teacher resources gives the definition of each as follows: data collection is the process of gathering appropriate information. Data analysis is making sense of data, finding patterns, and drawing conclusions. Data representation is depicting and organizing data in appropriate graphs, charts, words, or images. Problem decomposition is breaking down tasks into smaller, manageable parts. Abstraction is reducing complexity to define the main idea. Algorithms and procedures is the series of ordered steps taken to solve a problem or achieve some end. Automation is having computers or machines do repetitive or tedious tasks. Simulation is the representation or model of a process. Simulation also involves running experiments using models. Parallelization is to organize resources to simultaneously carry out tasks to reach a common goal.

Park (2018) also released that there are two steps where teachers can use the 9 components of CT seperately. Teachers in the study used the first three CT components in students' forming concepts related to the topic they learn for understanding the problem on the basis of curriculum. The last 6 components of CT were used in students' applying concepts for producing solutions on the basis of curriculum revised. Teachers were found to extend the latter part of the unit where students have chances to design the overhang of the roof of a Korean house by considering the Sun's light incident angle in different seasons (Hwang, 2019). Park and Park (2018) also released that 9 components (generalization instead of parallelization) of CT can be found in STEAM programs from the lower elementary level to the high school level with those components. The relative dominating components usage of CT were withdrawn according to different school levels of STEAM program and those usages were changing from the lower and the higher levels in kinds and their frequencies. This implied that we can decide if there is CT component or not in the class teaching or science lesson.

Weintrop et al. (2016) developed their taxonomy through five steps. Step one was a literature review. Step two was an open-coding of thirty-two classroom activities. Steps three and four was a revision of the taxonomy and external reviews by math and science teachers. Finally step five was interviews with STEM practitioners. Their findings lead them to break CT down into four major categories and then into a total of twenty-two subset practices rather than individual components (Table 1).

One of differences between the two systems is that ISTE and CSTA define the components as skills while Weintrop et al. (2016) name the components as practices. This difference came about due to input from high school science and mathematics teachers as part of a workshop where they were asked their opinions on the taxonomy. The teachers suggested the change as practices are "... broader and more actionable" (Weintrop et al., 2016). This is in line with the general trends in both science and mathematics

Table 1. Categories of computational thinking practice (Weintrop et al., 2016)

Data Practices	Modeling & Simulation Practices	Computational Problem Solving Practices	Systems Thinking Practices
Collecting Data	Using Computational Models to Understand a Concept	Preparing Problems for Computational Solutions	Investigating a Complex System as a Whole
Creating Data	Using Computational Models to Find and Test Solutions	Programming	Understanding the Relationships within a System
Manipulating Data	Assessing Computational Models	Choosing Effective Computational Tools	Thinking in Levels
Analysis Data	Designing Computational Models	Assessing Different Approaches/ Solutions to a Problem	Communication Information about a System
Visualizing Data	Constructing Computational Models	Developing Modular Computational Solutions	Defining Systems and Managing Complexity
		Creating Computational Abstractions	
		Troubleshooting and Debugging	

Table 2. The Protocols of Computational Thinking Practice (Park and Hwang, 2017)

CT practice		CT protocol
1	Connecting Computing (CC)	CC-1) Use computing to facilitate exploration and the discovery of connections in information.
		CC-2) Use computer to process information to gain insight and knowledge.
		CC-3) Appropriately connect problems and potential algorithm.
2	Developing Computational Artifacts (DCA)	DCA-1) Use computing tools and techniques to create artifacts (creative expression).
		DCA-2) Develop an algorithm designed to be implemented to run on an computer.
3	Abstracting (Abs)	Abs-1) Develop an abstraction.
		Abs-2) Describe the combination of abstractions used to represent data.
		Abs-3) Use large data sets to explore and discover information and knowledge.
		Abs-4) Use multiple levels of abstraction in computation.
		Abs-5) Use abstraction to manage complexity in program.
4	Analyzing Problems and Artifacts (APA)	APA-1) Analyze the considerations involved in the computational manipulation of information.
		APA-2) Evaluate algorithms analytically and empirically.
		APA-3) Evaluate a program for correctness.
		APA-4) Employ appropriate mathematical and logical concepts in programming.
		APA-5) Analyze how characteristics of the internet and the systems built on it influence their use.
		APA-6) Analyze how computing affects communication, interaction, and cognition.
		APA-7) Analyze the beneficial and harmful effects of computing.
5	Communication and Collaborating (Co-Co)	Co-Co-1) Communicate insight and knowledge gained from using computer programs to process in information.
		Co-Co-2) Express an algorithm in an language.
		Co-Co-3) Explain characteristics of the internet and the systems built on it.
		Co-Co-4) Connect computing with innovations in other fields.
		Co-Co-5) Connect computing within economic, social, and cultural contexts.
		Co-Co-6) Collaborate to solve a problem using programming.

pedagogical thinking to highlight the need for knowledge not just skill (NGSS Lead States, 2013). The skills by the ISTA and CSTA can be observed and described individually. The CT practices suggested by Weintrop et al. (2016) consist of concrete behaviors; data practices, modeling & simulation practices, computational problems solving practices, and system thinking practices, which can be described integrally.

Park and Hwang (2017) illustrated what practices of CT can be taking place during the STEAM class (Table 2). If we apply the individual 9 components as suggested by Park and Park (2018) to see if there is CT or not, we can show the results of CT usage of which one and how much. However, we do not know how those components are interacting with each other in the context of forming and applying science concepts. Park and Hwang (2017) extracted five different practices with 23 protocols illustrating each practice, where we can recognize how those practices are interacting each other (Fig. 1).

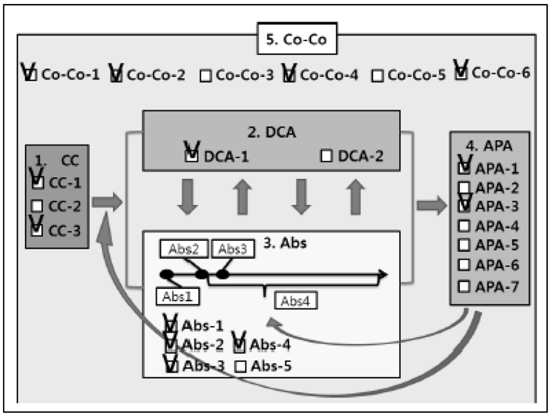

Fig. 1. CT practice checklist/flow in the context of STEAM (Park and Hwang, 2017)
CC (connecting computing)/DCA (developing computational artifacts)/Abs (abstracting)/APA (analyzing problems and artifacts)/Co-Co (communicating and collaborating)

In Fig. 1, Park and Hwang (2017) describes what kinds of CT practices students showed while performing their STEAM activities about climate change (middle school level) to find out the best solution. The authors described how science learning and teaching occur and how students formed and applied concepts which they learned during science lessons guided by a teacher. Students recognized the problem of climate change in the city where they live. Therefore, students had to decide what data were appropriate to indicate that the city also experienced climate change, how to collect the data, and how to transform these data to be represented. Students learnt concepts about climate change through experimentation and argumentations offered in the program and they also discussed and decided which way is the best solution to reduce CO_2 in the city where they live and where there are many chemical factories. Students designed the photo bioreactor to see how efficient this solution is. During these all processes, students had chances to experience all CT practices (Table 2), mainly CC, Abs, and Co-Co. The authors described that CC and Abs and Co-Co practices are more dominating than other two practices; APA and DCA.

The CT practices described in the STEAM program of climate change showed what and how CT practices are interacting each other (Fig. 1). CC practice had been initialized at the beginning of the program, and DCA and Abs interacted each other to be connected to APA. Co-Co practice covered all context of practices during the 10 lessons of climate change STEAM program. On the basis of those findings, the researchers in this study are trying to find out the relationship between the components of 9 (Park and Park, 2018) and the CT practices (Park and Hwang, 2017), which can be developed as a guideline with practices and components. If the researchers could make this guideline, it will be much easier for teachers to employ CT in the science classroom. This guideline can be used as planning and assessing tool for measuring CT skills.

CT is a catalyst for STEAM education: Over the past twenty to thirty years students' use of technology in the classroom (and people's everyday use) has increased drastically. Indeed, the term 'digital natives' is a growingly used term to describe the younger generations. It is as if young people grow up instinctively knowing how to use technology, to use social media to connect with friends, watch the latest music videos on the internet, and play online games. Does this knowledge really make them fluent in the use of technology, however? Could those same young people program a simulation of a container of gas molecules, or even create their own game? Coyle (2010) succinctly phrased it, "It's as if they can read but not write".

So, the question is why can these young people not 'write' when it comes to the use of technology? Margolis et al (2008) argues that these 'writing' skills are seen as (especially by females and people of color) as something for only the "best and the brightest". It is seen as something that they cannot do. It is not uncommon to hear people to claim that they don't have a science brain or that they are no good with numbers.

As discussed earlier about the purpose of science education in the 4th industrial revolution, STEAM subjects need to equip students with competencies to be literate scientifically through scientific inquiry. That has always been the case, but it is especially true now with the extensive use of computers in STEAM fields. The idea of the scientist of the future who does not know to use a computer to manage data and run simulations will seem as strange as a scientist who does not know algebra (Foster, 2006).

STEAM subjects need a way to solve both of these issues. The need to properly train and prepare students for their future in STEAM and to help keep the interest of students and encourage students from all backgrounds to actively participate in classes. The answer to both of these needs is bringing CT into the STEAM classroom. CT can provide the training that students will need in their future careers. It also makes sure that everyone is exposed to CT practices. When these practices are only taught in elective or after-school classes then there is an underrepresentation of females and minorities in those classes (Margolis and Fisher, 2003). They found that the computer science department of the Norwegian Institute of Technology had the lowest percentage of students, at only 8 percent, of all the different departments. They also found that in 1999 only 7 percent of Advanced Placement (AP) students in computer science classes were African American or Hispanic.

Park and Park (2018) released the possibility of including CT in STEAM program. The purpose of STEAM education is to equip students with competencies to be creative problem solvers and those programs have been developed to meet this goal. Therefore, students had chances to explore what problems they could face in their community and what solution would be best based on their evaluations. In this study, the implication of STEAM education had been withdrawn as follows; CT can promote students' opportunities of using technology as well as engineering disciplines for envisioned STEAM education. This study released the differentiated CT practices had been observed at different levels during a STEAM program, where simple CT practices had been observed at elementary levels and more advanced ones at high school level during climate change and water shortage STEAM lessons. Park and Park (2018) stated the result of this study can support the CT framework defined by NGSS (2012). On the basis of this study, the authors also emphasized the critical role of teacher education for computational thinking.

Weintrop et al. (2016) also illustrated the reciprocal relationship between CT and, science and mathematics. For example, of DNA sequencing from the ground up, students had chances to reassemble the songs related to the problem, which has been assessed by different approaches. Students also went on to more difficult challenges: reconstructing an unknown password with an unfamiliar combination of letters and numbers. Then they applied their technique to derive an efficient, robust, and general algorithm for sequencing this, which are the steps of preparing problems for computational solutions, creating computational abstracts. CT practice in this student work promotes students learning science. There is another example where high school students use an interactive simulation to explore the relationship between the macroscopic properties of gases based on how those properties emerge from microscopic interactions, which could make students build conceptual understandings. Students were observed to use practices of computational models to understand a concept. These are all cases of examples where students used CT practices named by the authors in this study.

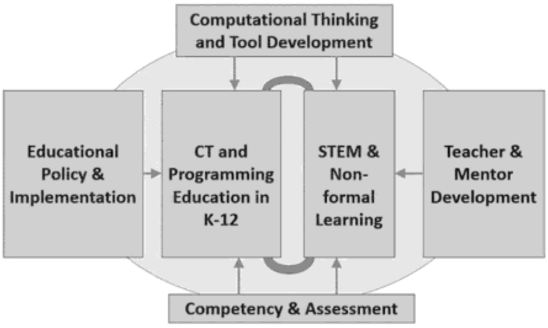

Fig. 2. Conceptual framework of computational thinking education (Kong et al., 2017)

All of the cases above explain how CT can be embedded into STEAM as well as mainly science lessons. Internationally some teachers are struggling for new educational policy about computational thinking. For this conceptual framework of computational thinking education, the more systematic interaction as well as connections are necessary for promoting STEAM education with CT practices (Fig. 2).

CT is a cognitive process to be learned:

"Computational Thinking will be a fundamental skill used by everyone in the world by the middle of the twenty-first century" (Wing, 2008). This means that CT is not just a skill to be used by experts in their fields or by students just looking to get a good grade on their test. It is used by everybody every day.

As computational thinking is such a crucial skill for everybody, it is important that students are taught about computational thinking and how to do it successfully. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability (Wing, 2006). Some examples of how the average person might think computationally in their everyday life might be deciding how to get to work that day, take the bus, a taxi, or drive the car. Trying to find out why the light isn't coming on, scanning the barcode of the groceries at the self-checkout line. Deciding the mode of transport to get to work is an example of performing an algorithm. The person needs to make decisions based on the time it would take to get there (are they late and need to be quick), how much would it cost (they only have a little change in their pocket), what time of day is it (rush hour or quiet roads)? The person would need to go through these questions and ask themselves what is the best mode of transport for them. Finding out why the light isn't working is problem decomposition. The person would break the problem down into the possible reason that the light is out. Has the bulb blown, a problem with the light switch, faulty wiring, has the fuse blown or is the power for the whole block out? When going through the self-checkout line and scanning the products the person is performing automation. The till will automatically scan and read the barcode, keep a tally of the cost, and accept payment.

The skills and practices of CT may also be of use in peoples' workplaces. For example, managers could evaluate the workings of the office to try and increase efficiency. Warehouse managers can track inventory to help with the shipping of orders and the need to purchase more stock. Government policy makers can also use CT to help inform policies. They could study large amounts of data on traffic to see if a change to the traffic light system could reduce the amount of car accidents. CT could be used to analyze voting data to determine the true will of the people.

Students at schools also need to be trained for this cognitive skill. Teachers need to provide certain questions for students' experience of CT skills. Without training in using CT skills, students cannot know how to face the problem, what data to collect, and what solution to produce. CT is a structured and proven method designed to identify problems regardless of age and computer literacy level. Students must learn how to decompose the given problem to be researchable. For example, teachers can ask, what is the problem? What factors can you find in the problem? What patterns can you find? CT users must be innovators. Teachers ask students to use different perspectives to determine what to extract from a problem in order to create a solution by continuous evaluations. What kinds of pros and cons can you find in this process or solution? How can you make this better economically? Why is this appropriate or not? By doing this training, students can leap from consumers to creators to meet the goal of science education, scientific literacy (Cummins, 2016; Hwang, 2019; Nardelli, 2019; Park, 2018). There are many reports and discussions regarding CT practices stating as follows; there is not common definition of CT now among scholars but we all agree that CT is a pivotal skill for people to live in the 21st century, where people understand a problem and formulate a solution. For this, the CT cognitive skills must be experienced from elementary level to equip students with competencies to be creative problem solvers (Park and Park, 2018).

The Implication of Computational Thinking in Science Education

In this review, the recent national/international efforts have emphasized the importance of CT to prepare students to thrive in their overwhelming technological society to enable individual empowerment to tackle complex problems (NRC, 2011; Park and Park, 2018). Given that CT focuses on problem solving and engaging students in decomposing problems to be abstracted and designing the processes to be automated, it is pivotal that K to 12 science educators and policy makers explore ways to include CT into their practices and curriculum. CT is not new, but CT is in more demand in the classroom now so that teachers equip students with competencies of skills to be problem solvers. Teachers can connect this CT to what they are doing in the classroom now, this is the best approach. While research in this area of CT is relatively new, there are promising outcomes highlighting the positive impact of CT ideas to be embedded into the science learning and teaching context like science museums as well as classrooms. Participating elementary teachers in Hwang (2019) demonstrated their competencies of revising their curriculum and their PCK on the basis of their understandings about CT. Participating teachers' perception of CT is pretty underdeveloped at the beginning of PDP but they formed and changed their perception of CT as they interacted with science educators about CT throughout the PDP program of 1 year.

There is a limitation in CT implementation into science education. As seen in the section 'The definition of computational thinking in science education', there are a number of different ways to characterize the skills and practices of CT. This gives the first major problem of CT, the fact that the definition has not yet been concretely set. As Denning (2017) noted teachers and educational researchers are still having difficulty with three questions; *What is computational thinking? How do we measure students' computational abilities? And Is computational thinking good for everyone?* For anyone to effectively teach something they must first understand it themselves. If the teachers don't know what CT is then they will have great trouble instructing students. Not being able to answer these questions could lead to problems (Jones, 2011). Denning (2017) believes this vagueness is the result of trying to make the definition of CT fit for all disciplines and fields. He also says that overreaching claims have been made, such as, 'computational thinkers will be better problem solvers in all fields', and that 'computational thinking is a superior way of thinking'. Easterbrook (2014) is also concerned about the vagueness of the definition for CT, but he also brings up some other concerns. CT involves preparing problems that can be solved by algorithms. But what about problems that cannot inherently be solved by algorithms, for example, problems with a question of ethics, problems that are concerned with the values of a society, or problems about societal change. This can lead to students ignoring problems that might have these concerns and instead focusing on algorithmically solvable issues. It is perhaps no surprise that computer science majors were found to be the most anthropocentric (Mann et al, 2014).

In conclusion, we believe in modern times that CT is the key to moving students from knowledge acceptors to problem solvers (MOE, 2015; NRC, 2012; Park, 2018, Park and Park, 2018; Wing, 2008). Enhancing science teachers' understandings about CT and highlighting connections to their curriculum and practices in the classroom. For this, science educators with experienced teachers from science and technology as well as computer science areas can develop PDP through which teachers can face, discuss, struggle with, and become familiar with CT practices. Googling CT on the internet is good place to start and connect with a community of teachers who have similar teaching concerns internationally as well as nationally (Barr and Stephenson, 2011; Yadav, Hong, and Stephenson, 2016). Hwang (2019) added that participatory action research (MacDonald, 2012; PAR after this) way is efficient for PDP where science educators consult science teachers to improve their teaching on the basis of their understandings about CT. Participating teachers demonstrated the newly forming and changing concepts and practices about CT also. Well-developed PDP with the use of PAR can enable and promote teachers' understandings and practices about CT into the classroom. Explicit questioning strategies can be good starting point; what kinds of questions can be possible in each CT practice developed by Wing (2008), Weintrop et al. (2015), CSTA and ISTE (2011), or Park and Hwang (2017). From a pedagogical view, the use of CT can promote learning of science content (Weintrop et al., 2015; Wilensky, 1995; Wilensky and Resiman, 2006). Science learning must provide a meaningful context where computational thinking can be applied, which make students the practical practitioner envisioned for the 21st century. This paper is still open to the broad discussion about bringing computational thinking into science education classrooms with the pros and cons, but at least we need the concrete and explicit competencies essential to be problem solvers.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2013R1A1A1A05007849).

References

Aho, A. V., 2011, Computation and computational thinking. Ubiquity, an ACM Press, 2011(1), 1-8 p. DOI: 10.1145/1895419.1922682

Barr, V. and Stephenson, C., 2011, Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? ACM Inroads, 2(1), 48-54 p.

Coyle, P., 2010, Computational thinking and STEM education. Retrieved from https://peadarcoyle.com/2010/09/05/computational-thinking/

Cummins, K., 2015, Five reasons why computational thinking is an essential tool for teachers and students. Retrieved from edgalaxy.com (cool stuff for nerdy teachers) https://www.edgalaxy.com/journal/2016/5/25/five-reasons-why-computational-thinking-is-an-essential-tool-for-teachers-and-students

Denning, P. J., 2017, Remaining trouble spots with computational thinking. Communications of the ACM, 60(6), 33-39 p.

Easterbrook, S., 2014, From computational thinking to systems thinking: A conceptual toolkit for sustainability computing. Retrieved from https://www.atlantis-press.com/proceedings/ict4s-14/13446

Foster, I., 2006, 2020 computing: a two-way street to science's future. Nature, 440(7083), 419 p.

Han, N. and Nam, Y., 2018, The change of elementary science gifted students' perception about engineers and engineering practices through science and engineering integrated (SEI) lessons, Journal of Korean Earth Science and Society, 39(3), 275-290 p.

Hannasari, R., Harahap, M., and Sinulingga, K., 2017, Effect of scientific inquiry learning model using scientific concepts map and attitudes to skills process science students. Journal of Education and Practice, 8(21), 48-52 p.

Hwang, K., 2019, Exploring elementary teachers' perception about computational thinking included science program during professional development program. Master thesis at Chosun University.

Jones, E., 2011, The trouble with computational thinking. Retrieved from http://www.csta.acm.org/Curriculum/sub/CurrFiles/JonesCTOnePager.pdf

Kong, S. C., Abelson, H., Sheldon, J., Lao, A., Tissenbaum, M., Lai, M., Lang, K., and Lao, N., 2017, Curriculum activities to foster primary school students' computational practices in block-based programming environments. In S. C. Kong, J. Sheldon & K. Y. Li (Eds.), Conference Proceedings of International Conference on Computational Thinking Education 2017 (pp. 84-89). Hong Kong: The Education University of Hong Kong.

Kuhn, D., 1986, Education for thinking. Teachers college Record, 87, 495-511 p.

Kuhn, D., 1993, Science as argument: Implications for teaching and learning scientific thinking. Science Education, 77(3), 319-337 p.

MacDonald, C., 2012, Understanding participatory action research: A qualitative research methodology option. Canadian Journal of Action Research, 13(2), 34-50 p.

Mann, S., Costello, K., Lopez, M., Lopez, D., and Smith, N., 2014, An ethical basis for sustainability in the worldviews of first year students, in Proceedings of the 2nd International Conference on ICT for Sustainability (ICT4S'2014). Atlantis Press.

Margolis, J., Estrella, R., Goode, J., Jellison Home, J., and Nao, K., 2008, Stuck in the shallow end: Education, race, and computing. Cambridge Mass: MIT Press.

Margolis J., and Fisher A., 2003, Unlocking the clubhouse: women in computing. Cambridge Mass: MIT Press.

Ministry of Education, 2015, The 2015 revised information curriculum.

Nardelli, E., 2019, Do we really need computational thinking?. Communications of the ACM, 62(02), 32-35p.

National Research Council, 2012, A framework for K-12 Science education, Washington. D.C: The National Academies Press.

National Research Council, 2000, Inquiry and the national science education standards. Washington, D.C., National Academy Press.

NGSS Lead States, 2013, Next Generation Science Standards: For states, by states. Washington, DC: The National Academy Press.

Osborne, J., 2010, Arguing to learn in science: The role of collaborative, critical discourse. Science (New York, N.Y.). 328. 463-466. 10.1126/science.1183944.

Park, Y-S., 2006, Theoretical study on the opportunity of scientific argumentation for implementing authentic scientific inquiry. Journal of Korean Earth Science Society, 27(4), 401-415 p.

Park, Y-S., 2010, Secondary beginning teachers' views of scientific inquiry: With the view of hands-on, minds-on, and hearts-on. Journal of the Korean Earth Science Society, 31(7), 798-812 p.

Park, Y-S., 2013, STEAM project Climate Change, program book funded by KOFAC (Korean Foundation for the Advancement of Science & Creativity).

Park, Y-S., and Hwang, J., 2017, The preliminary study of developing computational thinking practice analysis tool and its implication, Journal of Korean Society of Earth Science Education, 10(2), 140-160 p.

Park, Y-S., May, 2018, The study of elementary science program with computational thinking practices and its understandings by elementary teachers, The 1st Korean Geoscience Union conference 2018, Hong-Cheon, Kangwon-Do, Korea.

Park, Y-S. and Park, M., 2018, Exploring students' competencies to be creative problem solvers with computational thinking practices. Journal of the Korean Earth Science Society, 39(4), 388-400 p.

Sim, J., Lee, Y., and Kim. H-K, 2015, Understanding STEM, STEAM education, and addressing the issues facing STEAM in the Korean context, Journal of the Korean Association for Science Education, 35(4), 709-723 p.

The International Society for Technology in Education, 2011, Computational Thinking; teacher resoucres second edition.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., and Wilensky, U., 2016, Defining Computational Thinking for Mathematics and Science Classrooms. Journal of Science Education and Technology, 25(1), 127-147 p.

Wing, J. M., 2006, Computational thinking. Communications of the ACM, 49(3), 33-36 p.

Wing, J. M., 2008, Computational Tinking and Thinking About Computing. Philosophical Transactions of the Royal Society, 366, 3717-3725 p.

Wing, J. M., 2010, Computational thinking: What and why?. Retrieved from https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Wilensky, U., 1995, Paradox, programming, and learning probability: a case study in a connected mathematics framework. Journal of Mathematical Behavior, 14(2), 253-280 p.

Wilensky, U., and Reisman K, 2006, Thinking like a wolf, a sheep, or a firefly: learning biology through constructing and testing computational theories-an embodied modeling approach. Cognition and Instruction, 24(2), 171-209 p.

Yadva, A., Hong, H., and Stephenson, C., 2016, Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. TechTrends, 60(6), 565-568 p.

Manuscript received: July 18, 2019
Revised manuscript received: August 23, 2019
Manuscript accepted: August 23, 2019