International Journal of Technology and Design Education (2023) 33:863–881
https://doi.org/10.1007/s10798-022-09750-5

Bringing computational thinking to technology education classrooms: Hacking car activity for middle schools in the republic of korea

Woongbin Park[1] · Hyuksoo Kwon[2]

Accepted: 22 March 2022 / Published online: 8 August 2022

Abstract

The purpose of this study was to develop a program that incorporates computational thinking into technology education classrooms and to investigate its effect on students. Software (SW) education and physical computing education are frequently addressed topics in technology education, but education about computational thinking (CT) lacks interest and research. Therefore, it is necessary to further develop educational programs in technology. In this study, we developed a program integrating CT, which centered on technological problem-solving processes. The program comprised 12 total hours of hacking a remote control (RC) car using Micro:bit development tool. This study investigates the effects of the developed program with a single group pre- and post-test quasi-experimental design. Nineteen students participated in the study, completing survey instruments that measure CT competency and attitudes toward CT and technology, answering an open-ended questionnaire, and voluntarily took part in semi-structured interviews. The results showed that the technological problem-solving program positively affected participants' CT-related competencies. Moreover, we observed improvement in participants' attitudes toward technology due to the integration of CT into their technology education classes. This study provides a strong case for incorporating CT into technology education. It also suggests future research direction regarding the development of students' CT competencies in various technological problem-solving contexts.

Keywords Computational thinking · Attitudes toward technology · Problem solving · Physical computing · Technology education

✉ Hyuksoo Kwon Ph.D.
hskwon@kongju.ac.kr

[1] Gajaeul Middle School, Seoul, Republic of Korea

[2] Department of Technology & Home-economics Education, Kongju National University, #102 Human Ecology Building, Gongjudaehak-ro 56, 32588 Gongju-si, Chungcheongnam-do, Republic of Korea

Introduction

Software education is called by various names, such as “software (SW) education,” “programming education,” “coding education,” “computational thinking (CT) education,” and “computer science (CS) education.” This suggests that the government, corporations, civic groups, and educational community are all attempting to improve students’ computer-related capabilities (Kim & Lee, 2013). Recent efforts have shifted the focus from computerized information and communication technology (ICT) education to SW, programming, and competency-centered CT education (Kim & Lee, 2013). The term “computational thinking” was first used by Papert (1980) and evolved rapidly following Wing’s (2006) study. Computer science subjects apply computational thinking through components proposed by the International Society for Technology in Education & Computer Science Teachers Association (ISTE & CSTA, 2011). Additionally, the International Technology and Engineering Educators Association (ITEEA) created new standards called “Standards for Technological and Engineering Literacy” (STEL), which emphasized computational competency (International Technology and Engineering Educators Association, 2020).

Since 2019, each ministry in South Korea, including the Ministry of Science and ICT (2019), has actively promoted policies related to artificial intelligence (AI) in both education and industry. The Ministry of Education (2020) recently added subjects such as “Programming,” “Basics of Artificial Intelligence,” “Utilizing Artificial Intelligence,” and “Ethics of Artificial Intelligence” to the 2022 revised curriculum. Given these influences, the number of studies related to computational thinking in South Korea increased from 18 to 2015 to 66 in 2018 (Lee, 2019). In addition, research related to AI in South Korea, which did not emerge at all until 2019, strongly impacted SW-related research trends and immediately became the most popular education research topic in 2020 (Min & Shim, 2021). On the other hand, Lee (2021) found that 65.4% of elementary school teachers in charge of AI education report insufficient knowledge of how to apply AI convergence education to their subjects, demonstrating that teachers lack confidence in this area.

The proportion of secondary technology education within national curricular subjects and extra-curricular activities related to SW is less than 2% in a recent study (Lee, 2018). In other words, there is a perception that SW education is naturally included as part of the information subject in Korean national curriculum. On the other hand, according to Wing (2006), CT is cross-curricular, and computer-like thinking benefits all subjects. Therefore, because it is a universally necessary competency, research on CT is essential to improve the information and communication components of technology education. However, in the 2015 revised curriculum, the information subject was designated as an independent subject. The information and communication technology unit was split into separate communication technology and information technology subjects. The information technology unit is presented only in the information subject. As a result, research on and application of SW-related technology education has diminished. Thus, this study aimed to develop CT-based problem-solving activities that can be applied directly in school settings and to measure changes in technological attitude that result from these practices.

Literature Review.

Technology education in South Korea is undergoing many changes due to the influences of integrated education, maker education, SW education, and AI education. Arduino, a single board computer (SBC) capable of both interactive design and physical computing, is a learning tool that is frequently used to promote CT (El-Abd, 2017). Micro:bit and Raspberry Pi are other popular tools (Lee et al. 2017). Practitioners and scholars are actively implementing education and research utilizing SBC from elementary school to high school, tailoring each learning tool to the specific subject. SBC serves as a tool for integrated education and interactive art in information, for data and programming, for solving scientific problems, and for creative purposes in art (O'Sullivan & Igoe, 2004).

While it would be interesting to introduce physical computing or interactive design into technology education for technological problem solving, but this is not a simple prospect. First, we must consider that classes using physical computing tools do not necessarily develop students' understanding of real-world technological activities (Cederqvist, 2020a; Ivarsson, 2003). Navigating this requires various educational studies along with in-service teachers providing appropriate guidance to students. However, there is insufficient research on technology education related to interactive design and physical computing. There is a need for a standardized draft curriculum and evaluation criteria for teaching computer science and CT that integrate technology and engineering (Asunda, 2018; Wright et al. 2012). On the other hand, technology/engineering and computer science are different; thus, an independent curriculum must apply CT principles that focus on solving problems in the field of technology (Buckler & Koperski, 2017). In this regard, it would be appropriate to teach technology as an integrated course, so long as the plan of study explores physical computing in various technological fields rather than focusing solely on CS-based coding and data.

Because technology education is separate from information education in South Korea, its teaching contents and approach are somewhat different. Information teachers focus on coding and data education, while technology teachers prefer physical computing. A survey of technology teachers demonstrated that they are aware of the necessity of SW education and generally believe that the best form of SW education is gleaned from physical computing such as Arduino or robot (Kim et al. 2015).

Various forms of technology education cover physical computing, with different development boards used for different purposes (Hodges et al. 2020; Tan et al. 2016). Among them, Arduino, which was developed for the Interaction Design Institute Ivrea (IDII) program at a graduate school in Italy, is a typical example, and the British Broadcasting Corporation's (BBC) Micro:bit is also used frequently. There is also Raspberry Pi, a highly extensible computer based on the Linux operating system which outperforms the two boards (Love, Tomlinson, & Dunn, 2016). In addition, Seymour Papert has strongly advocated for robot education. The Lego robot, which Papert and colleagues developed in the 1980s, is currently one of the most widely used tools that allows for the study of physical computing via robotics (Stager, 2016). Papert (1980) has discussed the use of computers to improve technological study within mathematics education, moving away from virtual reality to provide interactive, integrated education. Furthermore, a variety of robots other than Lego are being applied in technology education (Mubin et al. 2013).

The purpose of teaching physical computing, which is featured in most technology classes, is integrated education. Physical computing can be applied in combination with electronic and computer technologies (Love, Tomlinson, & Dunn, 2016). Additionally, depending on a teacher's lesson design, physical computing can be applied in various ways in manufacturing, transportation, experiments, and robotics. Physical computing is based on system design that utilizes all sensors and actuators, so it can be applied to practically all technological fields. As robots also constitute a system, teaching robotics through physical computing is also possible.

Despite its strong growth potential, current CT education is inadequate. If CT is incorporated into technology education successfully, it will not only provide software education for students sitting in front of a computer but also will offer an integrated education that develops and coordinates collaborative abilities and cognitive and operational skills (Lee et al. 2013). Therefore, this study attempted to develop a program that applies CT during problem-solving activities in the technology classroom. The software hacks the hardware of an electronic product (a RC car) and allows students to program and control its software by utilizing their technological knowledge. The hacking program used in this demonstration can be applied beyond RC cars to all electronic products that control motors.

Methods

Research Design

This study developed a CT program for middle school students based on the literature review. The program was implemented not as a national curriculum subject but rather as a creative activity program. The program included four total lessons over 12 h (3 h per lesson). Due to the COVID-19 pandemic, the entire course was conducted online through real-time video lessons. Additionally, this study administered a technological attitude test and computational thinking test to the 19 middle school students participating in the experiment. This study employed quasi-experimental single-group design with pre- and post-test. After the program, all participants completed post-test surveys and open-ended questionnaires, and a subset of students underwent semi-structured interviews.

Participants

The program was a middle school creative activity curriculum. Students who consented to participate did so for a total of four three-hour lessons. All students ($N = 19$; 14 boys and 5 girls) were in the first (13 years old) or second (14 years old) year of middle school. None of the students had taken any computer-related classes except for a practical subject class during elementary school, and they did not encounter any related content in their technology class.

Program Development

We applied Mager and Beach's (1967) three-step procedure of preparation-development-improvement to develop the program. As shown in Fig. 1, in the preparation stage, we examined CT-related research, analyzed the 2015 revised South Korean National Curriculum, and reviewed elements required for technology subjects, then we set the subjects and lesson formats. As described above, because the information technology unit is excluded from the current technology curriculum, CT-related elements are not overly included.

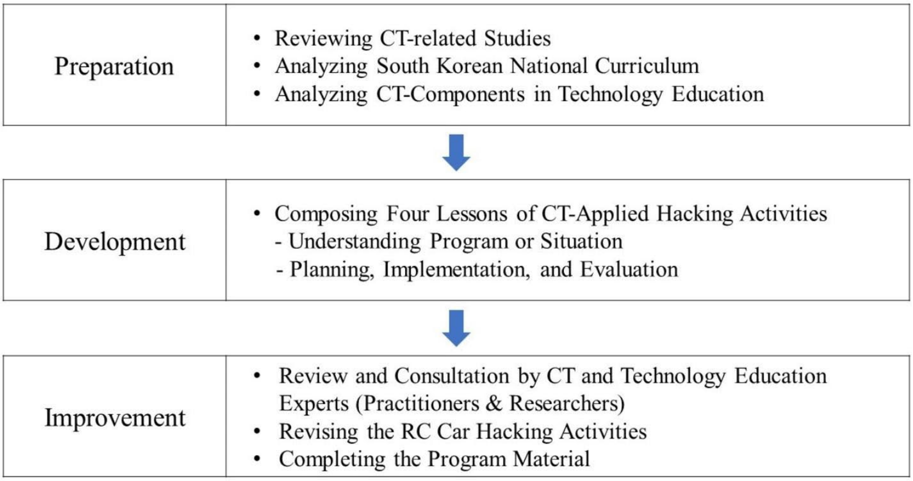

Preparation
• Reviewing CT-related Studies
• Analyzing South Korean National Curriculum
• Analyzing CT-Components in Technology Education
Development
• Composing Four Lessons of CT-Applied Hacking Activities
- Understanding Program or Situation
- Planning, Implementation, and Evaluation
Improvement
• Review and Consultation by CT and Technology Education Experts (Practitioners & Researchers)
• Revising the RC Car Hacking Activities
• Completing the Program Material

Fig. 1 Program Development Process

Therefore, it was necessary to develop a technology-centered CT program and analyze it to draw CT elements into technology education.

In the development stage, based on the curriculum analysis, we selected RC car hacking as the curriculum theme, which covers the electric vehicle elements of "transport technology," the communication medium of "communication technology," energy technology, and problem solving. We composed four lessons on CT-applied hacking, focusing on problem-solving activity steps (problem understanding, planning, execution, and evaluation) in accordance with the technology curriculum.

During the improvement phase, the program was reviewed through expert consultations related to computer technology, modified accordingly, supplemented, and completed. All five experts had more than 10 years of experience as technology teachers: one technology teacher (Bachelor's) representative of the Seoul Technology Teacher Community, one scholar (Master's) from the Seoul Office of Education, two researchers (Master's) from the Seoul Office of Education, and one technology teacher (Ph.D.). Table 1 presents a summary of the expert consultation.

Following recommendations from three of the experts, we expanded the sharing activities in Lesson 4. One expert's racing recommendation was not reflected because there were overlapping parts with sharing activities. We also modified the student activity guide to create a portfolio of all processes.

Program Development

The whole program (four lessons) was reorganized based on the four stages of problem-solving activities most often used in school technology classes, with each step spanning one three-hour lesson, resulting in 12 total lesson hours. The program revolved around the theme of hacking (upcycling) and asked learners how they would add new features (including electronic ones) if they had a broken toy. However, the students did not have broken

Table 1 Summary of Expert Consulting

Expert	Consultation
Expert 1	At the end of Lesson 4, I want to separate sharing and improvement, and improvement comes after self-evaluation.
Expert 2	At the end of Lesson 4, I wish the sharing activities could be expanded further.
Expert 3	Since the students have made cars, it may be possible to share their work or try a car race competition.
Expert 4	The role of the teacher is well defined, but the student activities should be presented in more detail.
Expert 5	It would be a good idea to guide students to record a portfolio throughout the entire course. Guidance on the degree of freedom of the students in the field of tool exploration is necessary, and it should be clear whether they are exploring the tool itself or exploring the car.

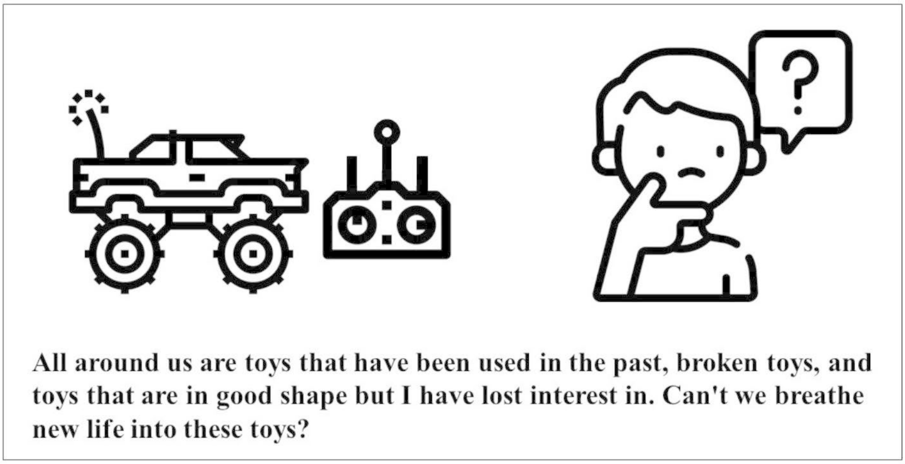
toys, so we decided to provide RC cars to all students. The students assumed that the toy was broken and began exploring ideas and designs for improving the car. Students created their portfolios on the website starting with the “Let’s Hack—Understanding the Problem” lesson. Students explored the mechanical elements of the toys and revitalized them via software that gave them new functions. After implementing physical computing and confirming that the toys worked as intended, all students had time to share problem-solving ideas. During and after every lesson, the teacher monitored the website on which the students expressed their thoughts, thus providing continuous feedback. Due to the COVID-19 pandemic, this study was conducted as a real-time, interactive individual project-type practice session. One teacher (a researcher) presented the lessons. All study procedures received approval from the Institutional Review Board (IRB). The IRB application included procedures for handling problematic situations and obtaining participants’ prior consent.

Lesson 1: “Let’s hack—understanding the Problem”

During the first stage, the students encountered the problem shown in Fig. 2. Although this study provided RC cars to all students, this study framed the question to include cases where students had toys with which they did not actually play. After learning hardware and software hacking (or upcycling) principles, students generated ideas by examining various examples implemented using physical computing. The teacher gave the students time to think about what functions they could and would like to add to their toys. Additionally, students started assembling portfolios of all courses using an online tool.

Lesson 2: “Mechanical elements and physical Computing—Planning”

To hack their own products, the students needed an understanding of machine elements and physical computing. They completely disassembled their toys to understand the structure. The teacher guided the students regarding the position and function of the motor, the type of power supply, and the circuit configuration. Moreover, students learned Micro:bit as an SBC for adding new functions. They learned how to implement an interactive system (physical computing) through sensors such as illuminance sensors and accelerometers and outputs such as LEDs and motors (Fig. 3). The teacher provided learning materials for various sen-



All around us are toys that have been used in the past, broken toys, and toys that are in good shape but I have lost interest in. Can't we breathe new life into these toys?

Fig. 2 Problem Situation Statement

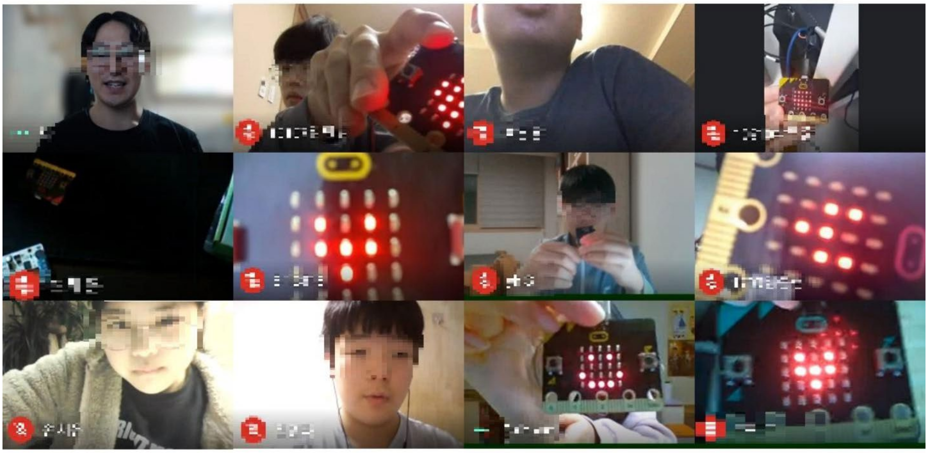


Fig. 3 Student's Activity Example 1: Creating Interactive Systems using Sensors and Actuators

sors and actuators, and the students individually studied the elements of physical computing explored in Lesson 1.

Lesson 3: "Adding New Functions—Execution"

After removing the Printed Circuit Board (PCB) that was basically attached to the product, the student connected Micro:bit to form a new circuit (Fig. 4). Using two of the original motors, students implemented programming that enabled forward and/or backward movement, left and/or right rotation, and stopping. Students could add new functions and different circuits depending on their abilities. The students modified their toys in various ways,

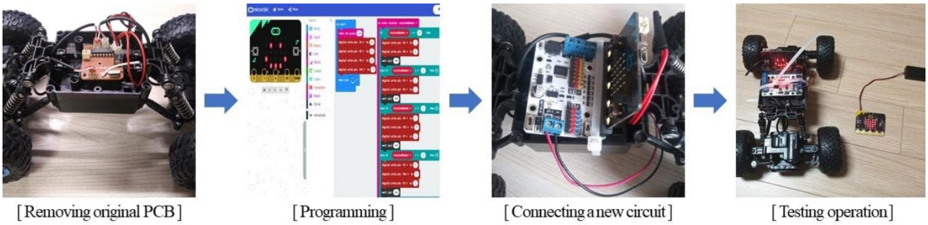

[Removing original PCB] [Programming] [Connecting a new circuit] [Testing operation]

Fig. 4 Procedure of Students: Hacking RC Car

such as changing the motor direction, increasing the number of control motors, changing the vehicle speed, adding LEDs, and recognizing tilt with an accelerometer. The teacher provided individualized guidance for each student.

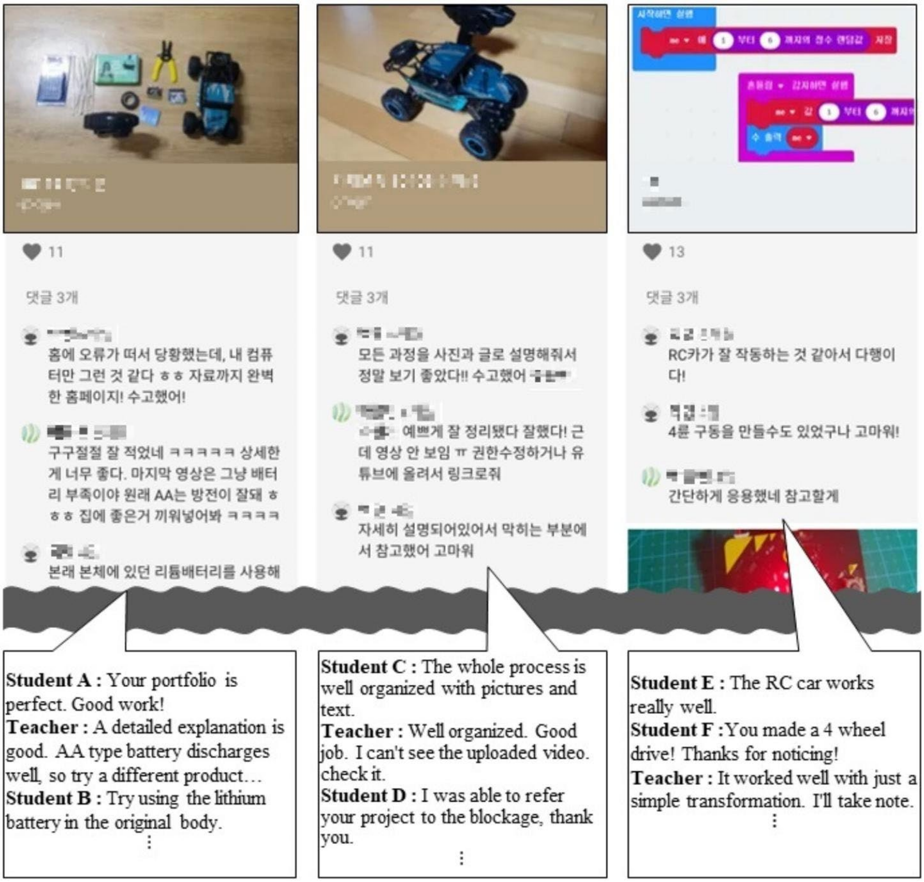
Lesson 4: completion and sharing Ideas—Assessment

Students recorded all lessons in their web portfolio. Students shared their know-how and the difficulties that they encountered in solving problems throughout their work. Additionally, they could freely consult and evaluate each other’s work through the idea sharing platform (Fig. 5). The teacher cultivated a constructive atmosphere and designed the presentation to be student-led.

Analytical Framework

After students completed all lessons, this study applied the single group pre- and post-test quasi-experimental design to investigate the program’s effects. As a mixed-methods study, quantitative measurement was conducted with two evaluations of technology attitude and computational thinking, before and after the program. For qualitative measurement, this study employed two open-ended questions on a post questionnaire for all 19 participants and conducted interviews with three participants who volunteered. This study conducted data analysis triangulation through the three methods: quantitative measurement, open-ended questions, and interviews (Leech & Onwuegbuzie, 2007). The pre-test occurred a few days in advance while we drafted the program agreement, and we prepared the post-test and open-ended questions for one day after completion of the program. We conducted interviews one week later, and the post-test took place at home (i.e., no teacher/researcher present) the day after the last class.

We piloted the quantitative test tools with 200 first-year middle school students to test reliability before administering them to study participants. The Pupils’ Attitude toward Technology (PATT) test consists of two constructs—interest in technology and results of technology—each comprising 10 items. Following the pilot study, we eliminated two items and three items from the two subscales, respectively, retaining 15 total items from this measure. The test measuring computational thinking comprises three constructs—computational thinking concepts, computational thinking performance, and computational thinking perspectives—measured via seven, four, and three items, respectively. We retained all 14 items for our study.



[redacted]

♥ 11

댓글 3개

[redacted]
홈에 오류가 떠서 당황했는데, 내 컴퓨터만 그런 것 같다 ㅎㅎ 자료까지 완벽한 홈페이지! 수고했어!

[redacted]
구구절절 잘 적었네 ㅋㅋㅋㅋㅋ 상세한게 너무 좋다. 마지막 영상은 그냥 배터리 부족이야 원래 AA는 방전이 잘돼 ㅎㅎㅎ 집에 좋은거 끼워넣어봐 ㅋㅋㅋㅋ

[redacted]
본래 본체에 있던 리튬배터리를 사용해

Student A : Your portfolio is perfect. Good work!
Teacher : A detailed explanation is good. AA type battery discharges well, so try a different product…
Student B : Try using the lithium battery in the original body.
⋮

[redacted]

♥ 11

댓글 3개

[redacted]
모든 과정을 사진과 글로 설명해줘서 정말 보기 좋았다!! 수고했어 [redacted]

[redacted]
[redacted] 예쁘게 잘 정리됐다 잘했다! 근데 영상 안 보임 ㅠ 권한수정하거나 유튜브에 올려서 링크로줘

[redacted]
자세히 설명되어있어서 막히는 부분에서 참고했어 고마워

Student C : The whole process is well organized with pictures and text.
Teacher : Well organized. Good job. I can't see the uploaded video. check it.
Student D : I was able to refer your project to the blockage, thank you.
⋮

[redacted]

♥ 13

댓글 3개

[redacted]
RC카가 잘 작동하는 것 같아서 다행이다!

[redacted]
4륜 구동을 만들수도 있었구나 고마워!

[redacted]
간단하게 응용했네 참고할게

Student E : The RC car works really well.
Student F : You made a 4 wheel drive! Thanks for noticing!
Teacher : It worked well with just a simple transformation. I'll take note.
⋮

Fig. 5 Students' Activity Example 2: Sharing Products and Interaction

The PATT instrument (Bame et al. 1993) was translated and implemented by Lee (2008) in South Korea. The instrument consists of six factors, with all responses measured on a 5-point Likert scale (Lee, 2008). Among the six factors, we measured only interest in technology and results of technology, which are most relevant to the program. Given the study's focus on technology subjects, we also measured students' technological attitude, which assesses the strategies used to approach technological problems.

To measure the effectiveness of computational thinking via robotics learning, we used a scale developed by Choi (2014) that examines computational thinking concepts (seven items), computational thinking performance (four items), and computational thinking perspectives (three items), with all responses utilizing a 4-point Likert scale. The scale assesses various concepts such as repeated actions, logic, variables, execution abilities (such as debugging and abstraction), and the abilities to express oneself and to cooperate.

In addition to these quantitative measures, we used an open-ended questionnaires and semi-structured interviews as qualitative measures. We asked students to explain in their own words what they learned from the program, posing questions such as "Summarize what you have learned while participating in the program" and "Explain in detail what you think

Table 2 Pre- and Post-test for Computational Thinking

Computational Thinking	Pre			Post			N	*p*
	M	SD	Mdn	M	SD	Mdn		
Computational Thinking Concepts	3.50	0.677	3.43	4.39	0.389	4.29	19	0.000
Computational Thinking Performance	3.09	0.791	3.00	4.03	0.571	4.00	19	0.000
Computational Thinking Perspectives	3.35	0.857	3.67	4.30	0.414	4.33	19	0.001

has improved or what you think you have learned through this program." Following the open-ended questions, we interviewed three of the 19 students. The interviews were semi-structured, with students responding freely for more than 10 min in a relaxed atmosphere. The main questions were "What did you learn from the class?"; "What competencies do you think you have developed?"; and "How did you solve problems when they occurred?" The researcher asked additional follow-up questions as needed.

Data Analysis

We conducted a normality test because the sample was small ($n < 30$), based on the central limit theorem (Kish, 1965). The Kolmogorov-Smirnov and Shapiro-Wilk tests produced significance values of less than 0.05 for some items. Accordingly, we performed a Wilcoxon signed-rank nonparametric test. Additionally, to complement the nonparametric tests and ensure our results' robustness, we performed data analysis triangulation. (Leech & Onwuegbuzie, 2007). We analyzed the open-ended question and interview responses to determine how much CT and technological literacy had improved. The CT analysis focused on elements defined in Choi's (2014) measurement tool, and the technological literacy analysis followed the definition in the PATT and the South Korean technology curriculum (Bame et al. 1993; Ministry of Education, 2015).

Results

Improvement in computational thinking

Computational thinking spanned three domains: concepts, performance, and perspective. The quantitative survey conducted before and after the program showed significant increases in average scores for all three areas (Table 2). Concepts scores improved from 3.50 to 4.39, performance scores improved from 3.09 to 4.03, and perspectives scores improved from 3.35 to 4.30. The Wilcoxon signed-rank test produced Z values of -3.662 ($p < .001$), -3.558 ($p < .001$), and −3.401 ($p = .001$) for concepts, performance, and perspectives, respectively.

Computational thinking concepts

Students identified that they understood concepts such as gyro sensors, buttons (sensors), and radio-wave communication. Additionally, the students learned how to use sensors and actuators simultaneously, and they indicated understanding computer science concepts such as variables.

Open-ended answers

Won: I learned that it can be operated directly through coding, or it is easy to work with multiple sensors.

Joo: I found it easy to manipulate with a gyro sensor (RC, etc.). I also found that other functions such as LEDs are built-in, so they can be easily used for other purposes.

Yun: I could interact with it by pressing a button (in Micro:bit), or it could be represented by an LED. I made the development board communicate with a radio wave system.

Gyuri: I used block coding, and I didn't understand variables well, but now I think I can use them well when I'm coding.

Computational thinking performance

Students searched for problems in the source code and solved the problem by modifying the code. Further, they showed that they could actively improve functions or control them as desired. In addition to the source code used in the program, their ability to utilize new programming by manipulating other source codes improved. One student who participated in the interviews (Yun) took the RC car home after the program ended. Afterward, he continued to actively engage with the software by modifying the source code to improve the car's functions.

Open-ended answers

Bum: I think I was able to learn how to solve these and other problems through coding.

Hun: It was amazing that I could program my own desired movements.

Ho: I used the source code to manipulate it myself, and I worked on fixing the parts that didn't work and improved the good things.

Won: I found that it was easy to code by using the code from the web.

Interview answers

Yun: It was my first time encountering such machines as a RC car and Micro:bit, so I didn't know how to operate or like […] Would I be interested in this? So, I tried different things with my dad with Micro:bit and also improved things. So, I think it was more meaningful.

Computational thinking perspectives

Students developed attitudes that allowed them to express themselves and explore the work as they intended from a computational thinking perspective. Another student who participated in the interviews, Son, was not familiar with programming; however, after participating in the program, she stated that she could identify and solve problems by herself rather than by imitating the teacher's actions.

Open-ended answers

Gyuri: I realized that with Micro:bit and expansion boards, I could make my own car.

Table 3 Pre- and Post-test for Attitudes toward Technology

Attitudes toward Technology	Pre			Post			N	*p*
	M	SD	Mdn	M	SD	Mdn		
Interest in Technology	4.22	0.670	4.50	4.44	0.518	4.50	19	0.008
Results of Technology	4.60	0.444	4.71	4.74	0.287	4.86	19	0.180

Gyu: I learned how to use coding to make RC cars or similar things move.

Interview answers

Son: The development board broke down in the middle. But, I was wondering if there was anything wrong, and I just took it apart from the beginning, so I think I got to know something better. […] At first, I saw tools such as Micro:bit for the first time, so I found out when I participated in the RC car program. It was really fun. […] When programming, I did something like Entry (South Korean block coding software) little by little, but if another teacher showed me, I was just following it. Now, I think of the block myself, move it, and make the LED appear so that I can understand the programming and get a perspective so I can picture in advance.

In summary, through the triangulation of objective survey items, open-ended questions, and interviews, we observed that students' relevant competencies increased in all three areas of CT.

Improvement in attitude toward technology

We measured attitude toward technology across two areas—interest in technology and results of technology—and observed slight improvements in both areas (Table 3). Scores for interest in technology increased from 4.22 to 4.44, and those for results of technology increased from 4.60 to 4.74. The Wilcoxon signed-rank test produced Z values of -2.656 ($p = .008$) and – 1.340 ($p = .001$) for interest in technology and results of technology, respectively.

Interest in Technology

We found that attitudes toward technology improved as negative perceptions of machinery disappeared through participation in the program. Additionally, we observed that students improved their problem-solving skills and developed more positive perceptions of technology through the activities. Technological literacy and interest improved through repeated step-by-step problem solving, especially considering that the technology program was one that participants had not completed previously. One participant, Son, explained that after participating in the program, her misconceptions of technology dissipated and her interest and literacy improved.

Open-ended answers

Ho: I changed my reluctance to use machines and my perception that technology was difficult. I want to participate in other programs as well.

Joo: In the past, when I disassembled something, I often tore it apart and broke it. It seems that I found hacking an RC car a little easier.

Bum: It seems like my ability to solve problems has improved. It seemed like it would require an expert to do things like dealing with electric wires, but I realized I could do it, too. Furthermore, I saw that technology could be easily accessible to anyone, regardless of age, as long as they are interested in technology.

Interview answers

Son: Before I came to school, I had no idea if I would like technology this way. When I hear about it, I just knocked on a hammer, and I thought about it; but, I didn't know how to do it, so I participated in a more fun way. I believe that my knowledge of technology has increased by manipulating Micro:bit and disassembling (the RC car).

Results of technology

Program participants' positive perceptions of technology intensified. They came away from the study recognizing that technology is beneficial and necessary to society. One interviewee initially felt that engineering was difficult; however, once he grasped the principles of motor control, he understood that technological literacy makes problem solving easier.

Open-ended answers

Hun: I hope that the school will make more of these programs so that other students will have a good time and experience these educational programs that benefit their social life.

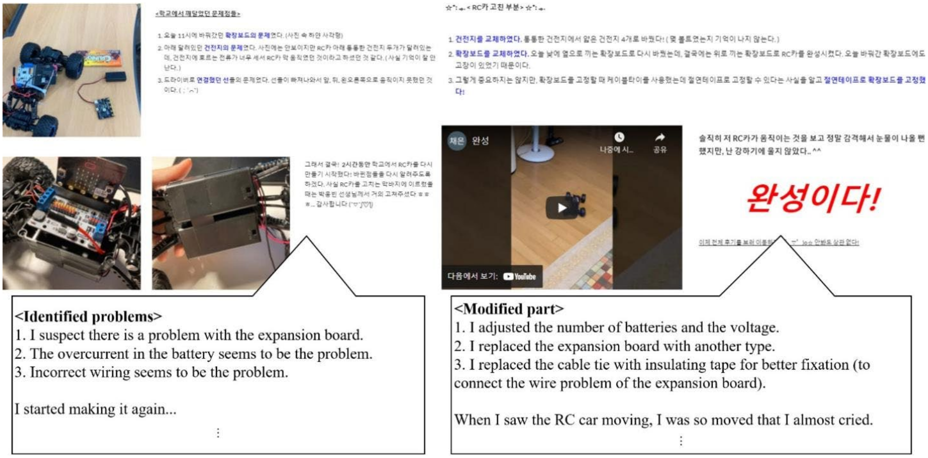
Gyu: I think it's a good thing because it brought me a step closer to technology. The future will surely require technology to live. Through this activity, my interest in technology has increased.

Interview answers

Joo: My technology teacher told me in class that if you have Micro:bit, you can now control any machine with any motor. I used to think that engineering was very difficult, but now that I know that I can control other motors with this, it has become so easy. In other classes, the other teacher taught me the definition of a motor, but my technology teacher just told me that this is a motor that runs when you connect these, and it was fun because it made me really understand.

Competency to Understand and solve problems

We observed a significant increase in students' problem-solving skills. Through a lot of trial and error, they developed a more persistent and resilient attitude toward facing challenges. One interviewee, Yun, responded that while participating in the program he exercised his individual cognitive skills frequently, which resulted in his ability to understand and solve problems independently. Additionally, students' problem-solving process manifested in their web portfolios (Fig. 6).



<학교에서 깨달았던 문제점들>

1. 오늘 11시에 바꿔갔던 확장보드의 문제였다. (사진 속 하얀 사각형)
2. 아래 달려있던 건전지의 문제였다. 사진에는 안보이지만 RC카 아래 통통한 건전지 두개가 달려있는데, 건전지에 흐르는 전류가 너무 세서 RC카 막 움직였던 것이라고 하셨던 것 같다. (사실 기억이 잘 안난다.)
3. 드라이버로 연결했던 선들의 문제였다. 선들이 빠져나와서 앞, 뒤, 왼오른쪽으로 움직이지 못했던 것이다. (; ´へ`)

그래서 결국! 2시간동안 학교에서 RC카를 다시 만들기 시작했다! 바뀐점들을 다시 알려주도록 하겠다. 사실 RC카를 고치는 막바지에 이르렀을 때는 박용빈 선생님께서 거의 고쳐주셨다 ㅎㅎㅎ... 감사합니다 (¯▽¯)♡

☆*: .｡. < RC카 고친 부분 > ☆*: .｡.

1. 건전지를 교체하였다. 통통한 건전지에서 얇은 건전지 4개로 바꿨다! (몇 볼트였는지 기억이 나지 않는다.)
2. 확장보드를 교체하였다. 오늘 낮에 옆으로 끼는 확장보드로 다시 바꿨는데, 결국에는 위로 끼는 확장보드로 RC카를 완성시켰다. 오늘 바꿔간 확장보드에도 고장이 있었기 때문이다.
3. 그렇게 중요하지는 않지만, 확장보드를 고정할 때 케이블타이를 사용했는데 절연테이프로 고정할 수 있다는 사실을 알고 절연테이프로 확장보드를 고정했다!

솔직히 저 RC카가 움직이는 것을 보고 정말 감격해서 눈물이 나올 뻔 했지만, 난 강하기에 울지 않았다.. ^^

완성이다!

이제 전체 후기를 보러 이동해 ...)o☆ 안봐도 상관 없다!

<Identified problems>
1. I suspect there is a problem with the expansion board.
2. The overcurrent in the battery seems to be the problem.
3. Incorrect wiring seems to be the problem.

I started making it again...
⋮

<Modified part>
1. I adjusted the number of batteries and the voltage.
2. I replaced the expansion board with another type.
3. I replaced the cable tie with insulating tape for better fixation (to connect the wire problem of the expansion board).

When I saw the RC car moving, I was so moved that I almost cried.
⋮

Fig. 6 Student's Activity Example 3: Web Portfolio

Open-ended answers

Dain: It was definitely completed, but the car shook. I went to school to see the teacher to exchange parts. After replacing the expansion board and the battery, I tried to run it again, but the wire was connected incorrectly, and there was smoke. Thanks to the teacher for suggesting various solutions.

Ho: It seems that my problem-solving skills and calmness have improved as I continuously revise the work that does not function. I gained a lot of self-confidence, and it seemed that my tenacity to do everything to the end also increased.

Gyuri: I modified the circuit twice; nevertheless, the backward movement was not possible due to wiring problems. However, I'm glad it worked out in the end. There have been several production failures, and it seems that these have fostered persistence in solving the problem.

Song: I was able to experience making it myself. […] If something went wrong, I learned how to find out what went wrong and how to fix it.

Interview answers

Yun: In the past, if I think I've done something wrong, I immediately asked, "Why doesn't it work?" But now I don't just ask. I think I have the ability to find problems myself because I think a lot. I think I have the ability to solve problems on my own. For example, if I am wrong on a math problem, I don't immediately ask a math teacher. I think, "Why is it wrong?"

In sum, through the triangulation of objective survey items, open-ended questions, and interviews, we observed that students' problem-understanding and problem-solving skills improved through trial and error, bolstered by their increased positive attitude toward technology.

Discussion and implications

Based on the post-program analysis, we observed improvement in students' competencies and attitudes toward both CT and technology.

First, understanding of CT concepts, CT performance, and CT perspectives all were measured. On Choi's (2014) instrument, students demonstrated improvement in all CT areas, including concepts (conditional statements, loops, sensors, actuators, operations, variables, etc.), performance, and perspectives. Initially, students only had knowledge of block coding through physical computing, but the program broadened their understanding to include sensors and actuators. In addition, students used conditional statements and looping statements in the process of controlling the RC car. They also developed their understanding of system concepts in communication technology while simultaneously manipulating these new sensors and actuators. Students indicated that they were subsequently more easily able to understand and implement programming concepts, performing more confidently than before, which confirms that both their conceptual knowledge and performance improved. Although the participants already had experience with block coding, they responded that the concept and performance related to CT increased through the program. It can be inferred that the developed program significantly improved students' knowledge and performance. The performance area showed the largest increase (3.09 to 4.03). In the process of improving the RC car's functionality and solving related problems, the students naturally acquired CT competency by modifying the source code. One student was previously uninterested in machines and only appreciated SW. However, machine manipulation motivated him/her to improve the SW skills. The perspective area includes computational representation, collaboration, and inquiry. Students recognized that their ability to express their ideas and to implement them as desired had improved, as had their inquiry and problem-solving skills. According to the interviewees, they initially lacked confidence because their only experience was with copying and pasting coding. However, one student stated that he/she had developed "the habit of encountering problems, understanding problems, and solving problems" as a result of the program, indicating the ability to explore CT in considerable depth. However, due to COVID-19, it was difficult to observe active collaboration of the group other than via their exchange of online messages. Further, students responded that their ability to simulate coding in advance improved, and we confirmed that their inquiry skills and CT performance were also enhanced. Overall, we observed great advancement in students' performance, and the strengthening of conceptual and perspective-based competencies may have facilitated this. All students in South Korea enter middle school having had experience in block coding in a SW class in elementary school. With this middle school program, many students responded that their self-confidence and competency in CT were improved. Through this, it can be confirmed that classes using these computer related tools, as found in previous studies, do not necessarily bring about real changes in students (Grover & Basu, 2017; Ivarsson, 2003).

Second, as a technological problem-solving program, there were significant effects with respect to interest in technology and results of technology. Particularly, we confirmed that the degree of interest was very high, with average scores increasing from 4.22 to 4.44 and the topic being prominent in the open-ended question responses and interviews. South Korean students have a strong awareness of technology but relatively limited technology subject (Lee, 2015). Nevertheless, a high proportion of respondents indicated that their interest in the technology subject had increased. Participants expressed in the qualitative data that it was their first experience, but it was interesting. Additionally, the original perception was simply thinking of machines or electric wires. However, this program provided activities that combined RC toys and computing, which are students' favorite fields, through problem-solving activities. These programs aroused students' interest in technology. Specifically, students responded that after participating in the program, their awareness of technology and engineering had improved, and many expressed the opinion that technology and engineering were not as difficult as they had previously thought.

Furthermore, most students expressed that their "ability to understand and solve problems and their knowledge of technologies" increased considerably. Because this problem-solving program required practical trial and error, we observed improvement not only in students' attitude toward technology but also in their problem-solving skills.

Overall, students responded that their problem-solving skills and technological abilities in programming and CT were strengthened. This study's program is distinguished from other programs in several important ways.

First, the program motivates students to solve problems in a self-directed way by exploring themes in which they are interested. This educational method, which combines physical computing and the application of theoretical principles to real-world examples, has been developed and implemented in various fields of education (Grasel et al. 2010; Rubio et al. 2013). As consequence of this program, we found that students were more interested and engaged in their own initiatives.

Second, it is a technology-oriented CT problem-solving program that leverages technology education's characteristics to facilitate real-world problem solving and encourage trial and error.

This program starts by encouraging students to develop a full understanding of the problems with the product (RC car), moves on to identifying the technological problems that occur, then asks students to solve the problems, and finally provides students the experience of improving the product through SW and mechanical engineering approaches. Students underwent much trial and error to solve the problems, learning how to approach and address challenges and improve the situation with which they were presented. Thus, students naturally improved their problem-solving skills, and the self-directed experience led to greater competency development.

In previous studies, block coding was displayed as an advantage in terms of syntax, but students still had difficulty constructing and understanding blocks; therefore, additional educational strategies are needed for students' understanding (Grover & Basu, 2017; Ivarsson, 2003). To solve this problem, this program was designed to strengthen students' motivation first, and to enable learning by integrating CT elements through problem-solving experiences. In fact, our study's participants expressed that they had experience in block coding but did not understand it well; after participating in this program, their related skills improved.

The purpose of this study was to determine whether problem-solving centered CT classes influenced students' CT and technological attitude. To this end, unlike the previous SW approach, real problem-solving learning was provided by reinforcing engineering and technological elements. As a result, students' physical computing skills were diversified and strengthened, and the program positively impacted students' technological competency and attitude. This study shows that technology-centered CT programs can positively influence students' attitudes toward technology, CT, and problem-solving skills simultaneously. In addition, it suggests the practical possibility of a CT program for technology education and the necessity of research.

Limitations and Future Work

This study conducted a 12-hour program as a creative experience activity for volunteers from a middle school in the Republic of Korea, not a regular class. Nonparametric statistics had to be applied because the number of study participants was 19. Moreover, since the students participating in the program are volunteers, they are already highly motivated. The mean of attitude toward technology (interest and result) was high on the pre-test, and a relatively small change could be confirmed compared to CT. Not only South Korea's technology teachers, but also those in other countries recognize the need for software-related educational competency, but the teachers' current teaching capability and confidence is insufficient (Jamieson-Proctor & Finger, 2008; Peeraer & Van, 2011). Even in the middle school setting of this study, the lack of confidence in teachers and difference in competencies may limit the application.

The results, conclusions, and limitations of this study suggest the following.

First, a long-term study with an expanded population should be conducted. The number of student participants should be increased to ensure normality. In addition, for more reliable statistics, more quasi-experimental studies that include a control group should be conducted. Furthermore, the program also needs to be implemented long term within the curriculum of technology education rather than through non-regular programs.

Second, competent teachers are key to the success of a problem-solving-based CT curriculum. Technology in-service teachers experienced difficulties in SW related education (Kim et al. 2015; Lee, 2021; Vinnervik, 2020). In addition, teachers often lack confidence in ICT education (Jamieson-Proctor & Finger, 2008; Peeraer & Van, 2011). However, since technology teachers that are developing students' technological literacy, they must take responsibility and strive to strengthen their competencies. Further studies should be conducted for in-service and pre-service technology teachers regarding the implementation of CT education.

Third, the findings of this study should be connected to the research for the new national technology education curriculum in South Korea. In technology education, researchers are studying classes that apply technological problem-solving using physical computing or development boards (Cederqvist, 2020b; Chung et al. 2020; Love, Tomlinson, & Dunn, 2016; Qu & Fok, 2021). As such, CT centering on problem solving should be integrated, which is the essence of technology education (Asunda, 2018). As this study demonstrates, technology education's strong focus on physical computing means that it is a subject that can very easily be combined with CT and CS principles.

Conclusions

This study developed a problem-solving centered CT program for promoting students' technological literacy. Thus, in this study, we measured CT alongside technological literacy.

More specifically, we measured students' attitude toward technology and each element of CT using a mixed-methods approach, observing significant improvement in all areas. In addition, students experienced improvement in various problem-solving competencies (e.g., understanding, inquiry, trial and error, etc.). Further, students who had previous experience in block coding only and lacked technological confidence acquired deeper conceptual knowledge and strengthened their programming performance through this intervention.

Taken together, these results suggest that when students participate in a technological problem-solving program they improve their CT-related competencies. The CT program, which combines engineering and technology, is expected to be used not only for creative activity curriculum but also for free semester and club activities. Providing an adequate education requires developing a regular technological curriculum, continuing research on programs that integrate CT, and expanding professional training for teachers.

References

Asunda, P. (2018). Infusing Computer Science in Engineering and Technology Education. *The Journal of Technology Studies*, 44(1), 2–13

Bame, E., Dugger, W., de Vries, M., & McBee, J. (1993). Pupils' attitudes toward technology—PATT-USA. *The Journal of Technology Studies*, 19(1), 40–48

Buckler, C., & Koperski, K. (2017). is computer science compatible with technological literacy? *Technology and Engineering Teacher*, 77(4), 15–20

Cederqvist, A. (2020a). Pupils' ways of understanding programmed technological solutions when analysing structure and function. *Education and Information Technologies*, 25(2), 1039–1065. https://doi.org/10.1007/s10639-019-10006-4

Cederqvist, A. (2020b). An exploratory study of technological knowledge when pupils are designing a programmed technological solution using BBC Micro: bit. *International Journal of Technology and Design Education*, 1–27. https://doi.org/10.1007/s10798-020-09618-6

Choi, H. (2014). Development of evaluation tools to verify the effectiveness of robotic education: linking social and cultural contexts and computing thinking. *Journal of The Korean Association of information Education*, 18(4), 541–548

El-Abd, M. (2017). *A review of embedded systems education in the Arduino age: lessons learned and future directions.* 79–93

Grover, S., & Basu, S. (2017). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. *In Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*, 267–272

Hodges, S., Sentance, S., Finney, J., & Ball, T. (2020). Physical computing: A key element of modern computer science education. *Computer*, 53(4), 20–30

International Technology and Engineering Educators Association. (2020). *Standards for technological and engineering literacy: The role of technology and engineering in STEM education*. Reston, VA: Author

Ivarsson, J. (2003). Kids in Zen: Computer-supported learning environments and illusory intersubjectivity. *Education, Communication & Information*, 3(3), 383–402

Jamieson-Proctor, R., & Finger, G. (2008). ACT to improve ICT use for learning: a synthesis of studies of teacher confidence in using ICT in two Queensland schooling systems. *Australian educational computing*, 23(2), 12–18

Kim, H., & Lee, S. (2013). Analysis of foreign information (computer) curriculum status. *KERIS issue report*

Kim, J., Huh, H., Kim, Y., & Kim, S. (2015). Analysis of technology teachers' perceptions and educational needs of software education in middle and high school technology subjects. *The Korean Journal of Technology Education*, 15(3), 50–72

Kish, L. (1965). *Survey sampling*. New York: John Wiley and Sons, Inc

Lee, A. (2018). Domestic research trends analysis of software education. *The Journal of Educational Information and Media*, 24(2), 277–301

Lee, A. (2019). Domestic Research Trend Analysis of Computing Thinking. *The Journal of the Korea Contents Association*, 19(8), 214–223

Lee, C. (2008). The Development of Technology Attitude Scale for Korean Pupils. *Journal of Korean Practical Arts Education*, 14(2), 157–174

Lee, D., Lee, S., & Lee, Y. (2017). An Analysis of Research Trend about Devices and Programming Languages in Physical Computing Education. *Proceedings of the Korean Society of Computer Information Conference, 25*(2), 379–380

Lee, K., Sullivan, A., & Bers, M. (2013). Collaboration by design: Using robotics to foster social interaction in kindergarten. *Computers in the Schools*, 30(3), 271–281

Lee, S. (2021). Convergence Education of Elementary School Teachers and Pre-service Teachers. *Journal of Korean practical arts education*, 34(1), 1–17

Leech, N., & Onwuegbuzie, A. (2007). An array of qualitative data analysis tools: A call for data analysis triangulation. *School psychology quarterly*, 22(4), 557

Love, T., Tomlinson, J., & Dunn, D. (2016). The orange pi: Integrating programming through electronic technology. *Technology and Engineering Teacher*, 76(2), 24

Mager, R., & Beach, K. Jr. (1967). *Developing vocational instruction.*

Ministry of Education (2020). *Education Policy Direction and Core Tasks in the Age of Artificial Intelligence*. The press release of Ministy of Education, 2020.11.20

Ministy of Science and ICT (2019). *National Strategy for Artificial Intelligence*. The press release of Ministy of Science and ICT, 2019.12.17

Min, J., & Shim, J. (2021). Analysis of domestic secondary computer education research topics. *The Journal of Korean Association of Computer Education*, 24(1), 29–36

Mubin, O., Stevens, C. J., Shahid, S., Al Mahmud, A., & Dong, J. J. (2013). A review of the applicability of robots in education. *Journal of Technology in Education and Learning, 1*(209 – 0015), 13

O'Sullivan, D., & Igoe, T. (2004). *Physical computing: sensing and controlling the physical world with computers*. Boston: Thomson

Papert, S. (1980). *"Mindstorms" Children*. Computers and powerful ideas

Peeraer, J., & Van Petegem, P. (2011). ICT in teacher education in an emerging developing country: Vietnam's baseline situation at the start of 'The Year of ICT'. *Computers & Education*, 56(4), 974–982

Qu, J., & Fok, P. (2021). Cultivating students' computational thinking through student–robot interactions in robotics education.International Journal of Technology and Design Education,1–20

Stager, G. (2016). Seymour Papert (1928–2016). *Nature*, 537(7620), 308–308

Tan, M., Yang, Y., & Yu, P. (2016). The influence of the maker movement on engineering and technology education. *World Transactions on Engineering and Technology Education*, 14(1), 89–94

Vinnervik, P. (2020). Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education*, 1–30. https://doi.org/10.1007/s10798-020-09602-0

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

Wright, G., Rich, P., & Leatham, K. (2012). How programming fits with technology education curriculum. *Technology and Engineering Teacher*, 71(7), 3

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.