CLASSROOM INTERACTIONS AND STUDENTS’ LEARNING OUTCOMES AT A ROBOTICS SUMMER CAMP

by

QU, Jing Ru

A Thesis Submitted to

The Education University of Hong Kong

in Partial Fulfillment of the Requirement for

the Degree of Doctor of Education

Sept 2020

Statement of Originality

I, QU, Jing Ru, hereby declare that I am the sole author of the thesis and the material presented in this thesis is my original work except those indicated in the acknowledgments. I further declare that I have followed the University's policies and regulations on Academic Honesty, Copy Right and Plagiarism in writing the thesis and no material in this thesis has been published or submitted for a degree in this or other universities.

QU, Jing Ru

Sept 2020

Abstract

The previous research on robotics education (RE) focuses more on robotics technology itself rather than the way students learn in RE and emphasizes students' learning outcomes instead of their engagement process. As a result, students' dynamic and complicated interactions with teachers, peers, and robots are rarely touched. Despite numerous empirical studies concerning classroom interactions in different educational fields, the classroom interactions in RE remain under-researched. Therefore, this research investigates three questions: 1) What are the features of students' problem-solving and computational thinking in a robotics learning environment? 2) What are the changes in students' problem-solving (PS) competencies and computational thinking (CT) skills in a robotics learning environment? 3) What are the relationships between three classroom interactions and students' learning outcome development? The mixed-method research was conducted at a four-week RE summer camp in the mainland China, among which the methods of pre-/post-test, rubric scoring, classroom observation, and semi-structured interviews were adopted. A total of 32 primary school students (i.e., 10 girls and 22 boys) and one dedicated robotics teacher completed all the lessons and data collection procedures of the summer camp.

This research found several features of students' PS and CT in a robotics learning environment. As indicated by the research findings, students' PS competencies and CT skills were significantly enhanced during the summer camp of this research. Additionally, the positive correlations between the time spent in three types of classroom interactions and two learning outcomes were identified. Lastly, several classroom interactions that might be effective in students' learning outcome development were characterized. Based on the findings above, this research conducted some discussions and proposed a new conceptual framework. Additionally, the contributions and implications for RE teaching and research were suggested.

Keywords: Robotics education, classroom interaction, mainland China

Acknowledgments

I would like to express my sincere gratitude to my principal supervisor, Dr. Fok Ping Kwan, for his continuous support of my Ed.D study. With great patience, critical insights, and immense knowledge, he had guided me in the research and writing of this thesis. Meanwhile, I appreciate his charismatic personality and profound knowledge and feel honoured to be his student.

I would like to specifically thank my co-supervisor, Dr. Yu Wai Ming, for her dedicated support and guidance. Having offered me great encouragement, Dr. Yu was always enthusiastic to assist in any way she could throughout my doctoral life. When I felt depressed, she gave me great comfort like my family.

I would like to thank Dr. Laitsch and Dr. Xu, the panel members, for giving critical comments and useful ideas for the improvement of this study.

I would like to thank every participant in this research and everyone who helped me with the data collection.

Special thanks go to my friends. Although I have no siblings, my friends cared and supported me just like my siblings.

I also want to thank myself for not being defeated by so many challenges in my doctoral journey.

Last but not the least, I would like to thank my beloved parents and husband who provided me with infinite support.

Table of Contents

List of Abbreviations

ACM	Association for Computing Machinery
CFA	Confirmatory Factor Analysis
CNKI	China National Knowledge Infrastructure
CSPSI	Chinese version of Shorter Problem-Solving Inventory
CT	Computational Thinking
CTR	Computational Thinking Rubric
CTY	Centre for Talented Youth
EdUHK	The Education University of Hong Kong
EFA	Exploratory Factor Analysis
ISTE	International Society for Technology in Education
PS	Problem Solving
PSI	Problem-Solving Inventory
RE	Robotics Education
RQ	Research Question
SPSI	Shorter Problem-Solving Inventory
S-R	Student-Robot
S-S	Student-Student
S-T	Student-Teacher

List of Figures

List of Tables

Chapter 1: Research Background

1.1 Introduction

Promotion of the robot industry is included in many developed countries' national strategies. China's robot research and development started in the 1970s (The State Council Information Office, 2016). In recent years, with the support of a series of policies, China's robot industry has developed rapidly. Although China has been actively promoting the development of the robot industry at the policy level, the talent shortage has become a crucial factor restricting the development of China's robot industry (Yang & Qiao, 2017). Therefore, the development of robotics education (RE) in primary and secondary schools so as to train qualified personnel for the future of China's manufacturing industry has become a major issue in China's basic education. With the rapid development of science and technology, robotics has become increasingly touchable and visible in our daily lives and even in the field of education (Benitti, 2012; Toh, Causo, Tzuo, Chen, & Yeo, 2016). Young children are inevitably exposed to the environment with numerous robotized devices. Many robotics curricula and activities within or outside of schools have appeared in China in recent years, and robotics curricula and activities within or off-campus are popular choices among students. However, two problems relating to this prosperity should be noticed. Firstly, many parents and teachers focus too much on students' visible outcomes, such as a successfully completed robot product, while some lifelong-learning competencies might be neglected. The second problem is that people pay too much attention to students' outcomes rather than the learning process, and, hence, it is difficult to determine what really happens during students' learning process and how students' learning is shaped.

Meanwhile, the application of educational robotics to cultivating students' 21st-century competencies has gained popular interest from education scholars (Eguchi, 2013). Nonetheless, previous research on RE has focused more on robotics technology itself rather than the way students learn in RE and emphasized students' learning outcomes rather than their engagement process, so that students' dynamic and complicated interactions with teachers, peers, and robots were rarely touched (Jung & Won, 2018). Although there were numerous empirical studies concerning classroom interactions in different educational fields, such as language education and science education (Chin, 2006; Kalu & Ali, 2004), classroom interactions in RE were still under-researched.

A number of previous studies showed abundant evidence regarding students' learning outcomes in RE, and many of them suggested that involving students in RE might develop their problem-solving (PS) competencies (e.g., Atmatzidou, Demetriadis, & Nika, 2018; Barak & Zadok, 2009; Castledine & Chalmers, 2011; Norton, McRobbie, & Ginns, 2007; Wagner, 1998) and computational thinking (CT) skills (e.g., Atmatzidou & Demetriadis, 2016; Bers, 2010; Bers, Flannery, Kazakoff, & Sullivan, 2014; Chen et al., 2017; Eguchi, 2014; Ioannou & Makridou, 2018; Lee et al., 2011; Leonard et al., 2016). However, few of them explained how students' learning was developed, and they fell to a *technological determinist paradigm* which "attributed the main cause of learning outcomes to the robotics technologies" (Jung & Won, 2018, p. 13). Therefore, RE researchers should explore, in more detail, students' learning process, such as classroom interactions, which might contribute to the development of students' learning outcomes.

To address the realistic and theoretical problems discussed above, this research achieved three main purposes. Firstly, this study explored the features of students' PS and CT in robotics learning. Secondly, the study evaluated students' PS competencies and CT skills in robotics learning, especially the possible changes of these two learning outcomes. Thirdly, the study identified the relationship between classroom interactions and students' PS competencies and CT skills.

At a four-week RE summer camp in the mainland of China, mixed-method research was conducted, adopting the methods of pre-/post-test, rubric scoring, classroom observation, and semi-structured interviews. A total of 32 primary school students (i.e., 10 girls and 22 boys) and one dedicated robotics teacher completed all the lessons and data collection procedures of the summer camp.

1.2 Background

Promotion of the robot industry was included in many developed countries' national strategies. In 2011, President Obama launched *Advanced Manufacturing Partnership* (AMP 2.0) and announced a commitment to investigating next-generation robotics (Office of the Press Secretary). In 2012, *Robotics and Autonomous Systems* (RAS) was identified by the British Government "as one of the Eight Great Technologies that support the UK Industrial Strategy driving efforts to rebalance the UK economy and creating jobs and growth" (RAS-

SIG Steering Group, 2014, p. 3), while *RAS 2020 the UK strategy* was proposed in 2014 as the first national strategy for promoting the robotics industry in the UK. Moreover, to start the robot revolution, Japan published the *New Robot Strategy* in 2015. The three pillars of Japan's robot revolution were summarized as: "Global base for robot innovation," "World's leading society maximizing robot capacity," and "World's leading strategy for a new robot era" (The Headquarters for Japan's Economic Revitalization, 2015, p. 10).

China's robot research and development started in the 1970s (The State Council Information Office, 2016). In recent years, with the support of a series of policies, China's robot industry has developed rapidly. In 2015, the State Council of China proposed *Made in China 2025*. It highlighted "high-end CNC machine tools and robots" as one of the key areas of vigorous promotion (The State Council of China, 2015, in section 3.6.2). Also, the key areas of the technological innovation roadmap clarified that the development focus of China's robot industry in the next 10 years is mainly in two directions. One is to develop serial products of industrial robots and key components, promote the industrialization and application of industrial robots, and meet the urgent need for the transformation and upgrading of the manufacturing industry. The second is to break through the key technologies of intelligent robots, develop a batch of intelligent robots, and actively respond to the challenges of a new round of technological revolution and industrial transformation (The State Council of China, 2015). Additionally, in 2016, the Ministry of Industry and Information Technology, National Development and Reform Commission, and the Ministry of Finance of China jointly issued the *Robot Industry Development Plan (2016-2020)*. Its main purpose was that, “through five-year efforts, a relatively complete robot industry system should be formed, technological innovation capability and international competitiveness should be significantly enhanced, product performance and quality should reach the international level, major breakthroughs should be made in key components, and the robot industry should basically meet market demand” (The State Council Information Office, 2016, in section 2.2).

Although China has been actively promoting the development of the robot industry at the policy level, the talent shortage has become a crucial factor restricting the development of China’s robot industry (Yang & Qiao, 2017). Therefore, the development of robotics education in primary and secondary schools so as to train qualified personnel for the future of China’s manufacturing industry has become a major issue in China’s basic education.

With the rapid development of science and technology, robotics has become increasingly touchable and visible in our daily lives and even in the field of education (Benitti, 2012; Toh et al., 2016). Young children are inevitably exposed to the environment with numerous robotized devices. Many robotics curricula and activities within or outside schools have appeared in China in recent years. Some native companies, such as Xiaomi and Huawei, have launched educational robot products to satisfy the demands of RE.

As can be seen from government documents, policymakers’ attitudes toward RE are also positive. For example, the *New Curriculum Standards for Senior High School (Information Technology)*, promulgated in 2017, listed “Brief Introduction to Artificial Intelligence” (p. 9) as a selective course (Ministry of Education, 2017a). Similarly, the *New*

Curriculum Standards for Senior High School (General Technology) listed "Designing and making the robot" (p. 33) as one selective core course "for helping students to deeply understand the relationship between human and robot, to explore the ideas and methods of hardware and software coordination, systematic control and path planning in designing and making robots; for enhancing the ability to integrate mechanical technique, electronic technique, and computer technology" (Ministry of Education, 2017c, p. 33). Additionally, in the *Next-generation Artificial Intelligence Development Plan* proposed by the State Council of China in 2017, the need to "set up artificial-intelligence-related courses in primary and secondary schools, and gradually promote programming education" (in section 5.6) was stated as one of the steps in the development of the Artificial Intelligence (AI) industry. Because creating intelligent robots was deemed as one significant part of AI industries (The State Council of China, 2017), involving students in creating robots is an important means by which K-12 schools can carry out AI education (Zhou & Ye, 2019). Consequently, to promote the development of the AI industry and education, RE has gained increasing support from students, parents, and educators in China, and has become an area with lots of potential.

One can see that there are many robot learning projects for children in China; however, two problems regarding this abundance should be noted. Firstly, people focus too much on students' visible outcomes, such as a successfully completed robot, while some lifelong-learning competencies, such as PS and CT skills, might be neglected. The second problem is that people pay too much attention to students' outcomes rather than the learning process;

hence, it is difficult to determine what really happens during students' learning process and how students' learning is shaped.

To address the first problem, this research evaluated students' learning outcomes based on their performances in PS and CT rather than successfully assembling a robot. RE is a new field which has not yet formed a mature knowledge system, so parents and teachers have a relatively shallow understanding of it. There are many parents who support RE mainly because it has a quick outcome: students may only need one lesson of study to create a machine that looks high-tech. It may result in a phenomenon that, some robotics teachers are more concerned on teaching the quick way to make a robot, rather than leading students acquire some transformable skills through learning robotics. Therefore, more research is needed to provide some evidence that RE has the potential of influencing students' future (such as cultivating their lifelong-learning skills), and this research adds to the body of related knowledge.

This research evaluated students' PS competencies and CT skills; however, different from previous studies, this research is process-oriented, explaining students' changes in PS and CT through analysing their classroom interactions. This also addresses the second problem identified previously. A number of previous studies showed abundant evidence regarding students' learning outcomes in RE, and many of them suggested that involving students in RE might develop their PS competencies (e.g., Atmatzidou, Demetriadis, & Nika, 2018; Barak & Zadok, 2009; Castledine & Chalmers, 2011; Norton, McRobbie, & Ginns, 2007; Wagner, 1998) and CT skills (e.g., Atmatzidou & Demetriadis, 2016; Bers, 2010; Bers, Flannery, Kazakoff, & Sullivan, 2014; Chen et al., 2017; Eguchi, 2014; Ioannou & Makridou, 2018; Lee et al., 2011; Leonard et al., 2016). However, few of them explained how students' learning was developed, and they fell to a *technological determinist paradigm* which "attributed the main cause of learning outcomes to the robotics technologies" (Jung & Won, 2018, p. 13). Therefore, to address this gap, this research not only evaluated students' changes in PS and CT, but also explained how these changes occurred from the perspective of classroom interaction.

1.3 Research Purposes and Questions

This study had three purposes. Firstly, this study sought to explore the features of students' PS and CT in robotics learning. Secondly, it sought to evaluate students' PS competencies and CT skills in robotics learning, especially the possible changes in these two learning outcomes. Thirdly, it sought to identify the relationship between classroom interactions and students' PS competencies and CT skills.

Accordingly, this study was guided by three research questions (RQs):

Research Question 1: What are the features of students' problem-solving and computational thinking in a robotics learning environment?

Research Question 2: What are the changes in students' problem-solving competencies and computational thinking skills in a robotics learning environment?

Research Question 3: What are the relationships between three classroom interactions (i.e., S-T, S-S, and S-R interactions) and students' learning outcome development (i.e., PS competencies and CT skills)?

These three RQs will be further elaborated in the chapter containing the methodology. Among the research questions above, RQ3 is the focus of this study, which means more effort was exerted and richer data were explored in answering this RQ. Also, the answer to this RQ was the main contribution of this research.

1.4 Organization of the Thesis

This thesis consists of six chapters. This chapter presented the introduction, background, and research purposes and questions of the research. Chapter 2 starts with a summary of the literature on RE, then reviews relevant RE studies on classroom interactions, PS competencies, and CT skills, as well as demonstrates the conceptual framework of this research. Chapter 3 introduces the research methodology of the study, which adopted a mixed-method research design including quantitative and qualitative methods. Chapters 4 and 5 present the quantitative results and qualitative findings, respectively. In the last chapter, research results and findings are discussed to answer each research question, and the revised conceptual framework is presented. Also, the contributions and implications of the results/findings and the limitations of this research are reported, as are the conclusions.

Chapter 2: Literature Review

2.1 Introduction

In this chapter, firstly, relevant studies on robotics education (RE) will be reviewed in terms of the research trend, definitions, theoretical foundations, methodology issues, and research focuses, to obtain a holistic understanding of the research background of RE. Also, some implications from previous RE studies will be summarized and discussed. Then, RE studies on classroom interactions, problem-solving, and computational thinking, which directly relate to the current study's conceptual framework, will be critically reviewed, and research gaps will be discussed. Lastly, the conceptual framework of this study will be proposed, and the concepts included in the framework will be explicitly interpreted.

2.2 Robotics Education (RE)

Since this research posits itself in the field of robotics education (RE), it is necessary to provide an overview of existent research in this field. Next, the research trend, definitions, theoretical foundations, methodology issues, settings, and research focus of previous RE research will be discussed.

2.2.1 Research Trend

Recently, research on RE has become increasingly popular. As systematically reviewed by Xia and Zhong (2018), after 2013, several high-quality empirical studies emerged in this field and an increasing trend could be seen. This research attempted to search relevant publications in the databases of the ACM (Association for Computing Machinery) digital library, Google Scholar, and CNKI (China National Knowledge Infrastructure). The ACM digital library includes comprehensive literature focusing on the field of computing; some previous systematic reviews of RE were done with this database (e.g., Anwar, Bascou, Menekse, & Kardgar, 2019; Bascou & Menekse, 2016; Benitti, 2012). Google Scholar is known as a large-scale database, which is suitable for exploring the research trend. CNKI is the most authoritative academic database of mainland China, which allows us to determine the research trend of RE in the context of China.

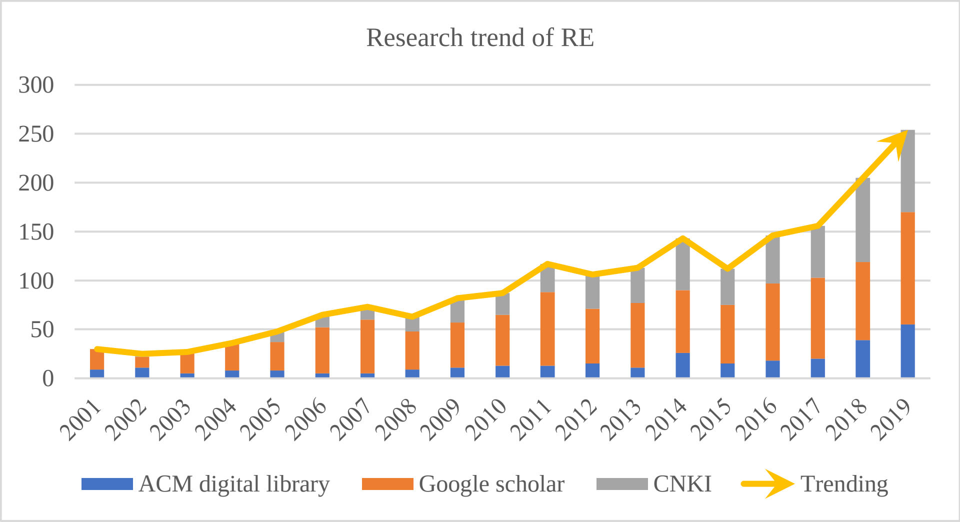


Figure 1. Research Trend of RE from 2001 to 2019.

Using the keywords of “robotics education” or “robot education” in both English and Chinese (i.e., 机器人教育) in the title, abstract, or keywords, as shown in Figure 1, from 2001 to 2019, research on RE showed an increasing trend in both a worldwide context and the mainland of China. Also, a remarkable increase could be identified since 2015, which indicates that, RE as a promising field of research, is getting increasing discussion.

2.2.2 Definition of RE

Properly defining RE is important. Before defining RE, it is necessary to clarify the position of robotics in the current study, as previous studies on RE “established different educational purposes for employing robotics” (Jung & Won, 2018, p. 4). As summarized by Jung and Won (2018), there were two perspectives in positioning robotics under the educational context. The first perspective posited robotics as an effective tool for teaching other subjects, while the second one “viewed robotics as a tool to teach robotics itself” (p. 5). In addition, Eguchi (2013) summarized previous studies by using three trends. The first trend posited “robotics as [a] learning objective” (p. 3), which aligns with the second perspective of Jung and Won (2018). The second trend viewed robotics as an aid to assist in teaching or helping students with special needs (Chang, Lee, Chao, Wang, & Chen, 2010). The third trend regarded robotics as a tool for enhancing students’ learning in different subjects (Toh et al., 2016), which echoes the first perspective of Jung and Won (2018).

The current study takes the second perspective of Jung and Won (2018) and positions itself on the first trend of Eguchi (2013); thereby, it defines **Robotics Education (RE)** as: *Using robot kits to form an environment for teaching robotics knowledge and nurturing learners' competencies/skills* (Benitti, 2012; Eguchi, 2012; Jung & Won, 2018). Specifically, this study regards robot kits as a tool in RE; the learning content of RE is robotics-intensified knowledge; and the purpose of RE is to nurture learners' competencies/skills.

Which should be noted is that, in previous studies, *Robotics Education* and *Robot Education* were alternatives, while *Robotics Education* was the most frequently used one. Also, compared to "*robot*" which is a tool applied and operated by the robotics area, "*robotics*" contains richer connotations, referring to "an area that deals with the design, construction, operation, and application of robots and robotic systems" (Jung & Won, 2018, p. 7). Moreover, as defined before, robotics knowledge is the main learning content of RE, while a robot is a tool through which students can acquire robotics knowledge and achieve higher-level skills. Thus, considering the factors above, the current study adopts Robotics Education rather than Robot Education.

2.2.3 Theoretical Foundations of Previous RE Research

Two dominant educational theories underpinned previous RE research: Piaget's constructivism and Papert's constructionism (Anwar et al., 2019; Eguchi, 2012; Jung & Won, 2018; Mubin, Stevens, Shahid, Mahmud, & Dong, 2013; Wong, Chew, & Wong, 2016). Both theories especially contributed to the pedagogical design of RE (Mubin et al., 2013). Piaget's constructivism is the most fundamental theory in RE, which regards knowledge as an experience actively constructed by children's interactions with their world (Eguchi, 2012; Jung & Won, 2018). Therefore, children need an object to put their hands on and think with (Ackermann, 2001; Eguchi, 2012). Also, children have their own views of the world. They interpret and internalize information from outside, using their own methods to relate new information to their prior knowledge and experience (Eguchi, 2012); their views could be coherent and robust (Ackermann, 2001). Children are stubborn, so they will not easily abandon their views, even if someone else tells them they are wrong (Ackermann, 2001, 2004; Eguchi, 2012). Hence, as facilitators, RE educators maintain the task of "offer[ing] opportunities for children to engage in hands-on explorations that fuel the constructive process" (Ackermann, 2001, pp. 1-2).

Mubin et al. (2013) pointed out that, in the RE field, there was a gradual shift from Piaget's constructivism theory to the modern constructionism theory suggested by Papert. This might be because constructionism theory is well fitted to the RE area, as it asserts that learning occurs when children are "engaged in designing and building their own personally meaningful artifacts and sharing them with others in a community" (Bers, 2008, p. 125), which focuses on *learning by design* (Bers, 2010). Meanwhile, robots are naturally tangible and should be physically manipulated as part of the learning process (Mubin et al., 2013). Constructionism was developed from constructivism, so they are both *constructivists* in that they regard children as the constructors of their knowledge and world, and both *developmentalists* in that they agree with the incremental perspective of knowledge construction (Ackermann, 2001). Therefore, Papert also stressed “nurturing a process of development in the child with its own internal coherence” (Papert, 2001, p. 3). Nevertheless, constructionism expands constructivism by paying “particular attention to the role of constructions in the world as a support for those in the head” (Bers, 2008, p. 125). The difference between constructivism and constructionism is that:

> Papert’s research focuses on how knowledge is formed and transformed within specific contexts, shaped and expressed through different media, and processed in different people’s minds. While Piaget liked to describe the genesis of internal mental stability in terms of successive plateaus of equilibrium, Papert is interested in the dynamics of change. (Ackermann, 2001, p. 8)

Based on constructionism, previous studies took advantage of the tangibility and functional properties of robot kits to address “the appropriateness of robotics education for young children” (Jung & Won, 2018). Also, in a constructionism RE environment, children learn more effectively when less instruction or teaching is involved in the learning process (Eguchi, 2012). Thereby, students could learn best “by making, creating, programming, discovering and designing their own object to think with in a playful manner” (Bers, 2008).

However, Jung and Won (2018) argued that, although the adoption of constructivism and constructionism theories in RE teaching and learning was useful, most of the past studies narrowed down these theories into a *technological determinist paradigm*. In the educational context, the technological determinist paradigm regards robot kits and robotics technologies as the main cause of students' learning outcomes; however, it simplified the interaction between children and robot kits as unidirectional rather than bidirectional or multidirectional (Jung & Won, 2018). An example of this determinist paradigm is in Mubin et al. (2013), which viewed robot kits as having a passive role and ignored the bidirectional/multidirectional interactions between people and technology. Due to this kind of determinist paradigm, previous studies paid more attention to the casual influence of implementing robot kits (i.e., such as Lego Mindstorms) to students' learning outcome measures (Jung & Won, 2018), while they failed to address the principle of constructivism and constructionism, which suggests that we should focus on students' interactions with their environments and learning process. Moreover, with regard to the situation that the main body of current RE research is based on constructivism and constructionism perspectives, Jung and Won (2018) suggested adopting social and cultural frameworks in the RE field, because this may help us know "how young children make personal, social, and cultural meanings from robotics kits and in what ways young children interact with the robotics kits beyond prescribed and expected actions" (p. 15).

2.2.4 Methodology Issues of Previous RE Studies

According to Xia and Zhong's (2018) systematic review, most of the previous studies they reviewed were conducted with primary school students (57.14%), with a small sample size containing fewer than 40 participants (47.62%) and with a short duration of less than four weeks (37.5%). This suggests that relevant studies focused more on primary school students. The reason for this situation might be that robotics' tangible nature shows its potential for engaging young children in STEM-related learning (Bers, 2008, 2010; Bers et al., 2014; Chen et al., 2017; Jeschke, Kato, & Knipping, 2008). The reason why several previous RE studies contain a small sample size (fewer than 40) might be because of the limited availability of robotics kits, as robotics kits and computers are relatively expensive. For example, in the current study, the summer camp could afford and provide only 10 sets of robot kits and 10 laptops for only 10 students; therefore, each class must be small to guarantee that every student has an opportunity to use the robot kits and computers. In addition, because robotics activities allow students to explore freely, and require a teacher to pay a significant amount of attention to children's operational behaviours with robot kits and computers, a small class size could be easier to organize. The short-term project is popular in the RE field, due to the advantages of the informal learning environment, as stated before. However, Mubin et al. (2013) remind us about the one-off problem of the short-term project, which might lead to questioning of students' longitudinal impact.

Most of the previous RE studies implied a non-experimental research design. In Xia and Zhong's (2018) review study, non-experimental design counted for 59.09% (13) of the studies they reviewed, while in Bascou and Menekse (2016), the percentage is 87.80% (104) and in Anwar et al. (2019) the percentage is 71.43% (105). Xia and Zhong (2018) also argued that some previous studies claimed to use an experimental design but did not randomly assign samples to groups. Nevertheless, Jung and Won (2018) reviewed high-quality studies in RE and proposed that most of the studies they reviewed were experimentally designed. However, what should be critically considered is that a non-experimental design is not necessarily weaker than quasi or pure experimental design in RE, as "we can gain lots of valuable information by a well-conducted non-experiment" (Xia & Zhong, 2018, p. 275).

As for data collection methods, the numbers of high-quality studies employing quantitative, qualitative, and mixed approaches were almost equal (Jung & Won, 2018). Specifically, observation, questionnaire, evaluation of artifacts, verbal interview, test/examination, neuropsychological test battery, and self-report were usually employed to evaluate students' learning performance in RE (Xia & Zhong, 2018).

2.2.5 Settings of Previous RE studies

Previous RE studies arranged robotics activities in either formal or informal learning environments (Bascou & Menekse, 2016). Nevertheless, RE is still widely deemed "an extra-curricular activity and a part of informal education" (Mubin et al., 2013, p. 5), especially at the K-12 level. This might be because informal sessions, such as after-school workshops and summer camps, have some advantages due to their short-term duration and minimum curriculum design and teacher training (Mubin et al., 2013). Also, robotics activities usually require an environment with specific hardware allowing children to use robot kits and computer programs, while the formal classroom environment could not provide such support, especially in the general context of mainland China (Zhang & Zhang, 2008). Moreover, for data collection issues, an informal environment could easily be set up as an experimental environment where environmental variables could be controlled, and participant recruitment might be more operable so that demographic variables could also be better controlled.

About the robot kits used in previous RE studies, in a review conducted by Mubin and colleagues (2013), a hypothetical progressive scale of the embodiment of the robot was raised, which classifying existing robot kits into four types based on the degree of embodiment: The mechanical kits with single function, the kits which have potential to teach both robotics and electronics (such as Arduino and Boebot), the robot kits which "allow more mechanical freedom and flexibility with the robot design" (Mubin et al., 2013, p. 4) (such as LEGO Mindstorms), and humanoid robots. Mubin and colleagues (2013) suggested that the selection of the type of robot kits should be based on the consideration of students' age, subject, and cost. As the pioneer of RE brand, LEGO has achieved great success in the past two decades. The majority of previous RE research was interested in LEGO-based activities, and LEGO is recommended for the age seven or older children (Benitti, 2012).

As regarding the programming interface used in RE, graphical programming interfaces, such as Robolab and Scratch, were popular among young age students (Olabe, Olabe, Basogain, & Castaño, 2010). It is because, comparing to text-based programming interface, such as C, C++, or JAVA, the visual feature of graphical programming is more attractive and accessible to children whose level of logical thinking is at a lower level (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010).

2.2.6 Research Focuses in RE: Students' Gains

In the early years of this century, due to the limited empirical evidence, researchers worried about the effectiveness of using robotics for enhancing students' learning (Benitti, 2012). Johnson (2003) pointed out that if there is no supporting evidence of students' gains in robotics learning, RE might simply become a *fashion*. Therefore, an increasing number of RE studies tended to focus on students' gains in terms of knowledge, skills, and attitudes (Jung & Won, 2018). Most of them found positive gains among students in RE, while some failed to find significant results.

Students' knowledge gains in RE. Previous RE studies often focused on knowledge of STEM-related domains. For example, in Hussain, Lindh, and Shukur's (2006) LEGO robotics training program, pre-/post-test results of control and training groups were analysed and a significant positive shift of Grade 5 students' mathematics performance was found in the training group (M_{pre} = .71, M_{post} = .92, p < .001). At a robotics summer camp for middle school students, Williams and colleagues (2007) found that participants' physics content knowledge significantly increased (M_{pre} = 8.40, M_{post} = 9.75, t(20) = -3.28, p = .004), as robotics activities provided opportunities for students to practice physics concepts such as the impacts of weight, angle, diameter, and friction on the vehicle's movement. In Church, Ford, Perova, and Rogers (2010), LEGO Mindstorms was used in the science curriculum. From its observation data, robotics provided rich opportunities to engage students in physics concept learning. For example, students used LEGO's rotation sensors to investigate the speed and acceleration of the vehicle and used sound sensors to explore sound waves. In Barak and Zadok (2009)'s robotics course for junior high school students, observation data suggested that project-based learning was useful in teaching students basic knowledge about scientific-technological concepts such as force, torque, and scientific units (such as Newton and Newton × meter). Barker and Ansorge (2007) examined primary students' comprehension level of SET (i.e., Science, Engineering, Technology) concepts after learning robotics. By comparing the control group and experimental group, they found a significant difference between the two groups' learning outcomes ($\mathrm{M}_{\text{control}}$ = 7.44, $\mathrm{M}_{\text{experimental}}$ = 17, t(22.17) = 12.93, p < .001). Additionally, as suggested by qualitative data, students' understanding of programming knowledge was found to be better.

Moreover, robotics knowledge was also deemed to be a crucial gain in previous studies. Although there was a lack of consensus regarding what should be included in robotics knowledge, at least two dimensions of robotics knowledge could be found in previous studies: knowledge related to constructing/building, such as understanding the elements of a robot and engineering principles, and knowledge related to programming, such as understanding commands and debugging concepts. Regarding constructing/building knowledge, for example, in Chambers, Carbonaro, and Murray (2008), students' understanding of the function of gears (i.e., the direction of turning, relative speed, the number of revolutions, and mechanical advantages) was evaluated by pre-/post-interview, and the results indicated that robotics learning helped develop students' understanding of gears' direction of turning, relative speed, and number of revolutions; most of the students were unable to explain the mechanical advantages of gears. Another example, in Sullivan and Bers's (2016) eight-week robotics curriculum for kindergarteners, the majority of students could correctly identify motor (97.56%), lantern (95.12%), distance sensor (90%), sound sensor (82.92%), and light sensor (70.73%), which indicated that participants had a good understanding of the elements of a robot. Regarding programming knowledge, for example, in Wang, LaCombe, and Vollstedt (2008), with the ROBOLAB programming environment, most of the students could demonstrate programming concepts including GoTo's (82.6%), conditionals (75.3%), loops (58.3%), nested structures (74.5%), variables (75.5%), functions/arguments (60.3%), and subroutines/subprograms (80.6%). In Sullivan's (2008) robotics summer camp project, called *Center for Talented Youth* (CTY), in the pre-/post-test, students were asked to answer open-ended questions about a robotics system by using programmatic terms; then, their answers were rated. Finally, the T-Test result indicated a positive significant increase from $M_{\text{pre}} = 23.09$ to $M_{\text{post}} = 25.82$ with $t(21) = 30.04$ and $p < .05$. Some other previous studies focused on both constructing/building knowledge and programming knowledge. For example, in Petre and Price's (2004) robotics program, by observation and interview methods, the researchers found direct evidence that robotics can effectively guide K-12 students in understanding engineering and programming principles such as how to design a robot step-by-step (*engineering principles*) and how to debug with the faulty program (*programming principles*). According to Jung and Won's (2018) review study, previous RE studies paid much attention to programming knowledge, while few talked about constructing/building knowledge. This might be because the robot kit's programable features and its advantages in terms of promoting coding education and computational thinking were increasingly recognized by the RE area (Bers et al., 2014; Chen et al., 2017). Nevertheless, because constructing/building a robot is also an integral part of RE, researchers should balance the two dimensions of robotics knowledge in RE research.

Students' skill gains in RE. In terms of the skill aspect, the focuses of previous RE studies were varied. Some concerned students' thinking skills. For example, Ricca and colleagues (2006) identified the growth of students' critical thinking in a LEGO Mindstorms program. However, they failed to provide specific statistics of the pre-/post-test. In Kazakoff and Bers' (2008) adapted TangibleK program, by scoring students' sequencing ability with sequencing cards, a significant increase ($t(33) = 2.71$, $p < .01$) from pre-test ($M_{\text{pre}} = 7.06$) to post-test ($M_{\text{post}} = 8.44$) was indicated.

Some focused on scientific skills. For example, in Williams and colleagues' (2007) robotics summer camp, the results of the pre-/post-test indicated that the increase in middle school students' scientific inquiry skills was not statistically significant ($M_{\text{pre}} = 3.50$, $M_{\text{post}} = 4.28$, $t(20) = -1.87$, $p = .077$), while qualitative data from teachers' reflective journals and interviews showed conflicting findings which supported students' skill enhancements. In Sullivan's (2008) CTY summer camp, students' skills of thinking and science process were coded by the frequencies of use. It found that the most frequently used skill was observation (30.73%), followed by the skills of evaluation of the solution (19.27%) and estimation (13.01%). The least-used skill was computation (1.30%). In AÇIŞLI's (2016) 20-hour robotics course, students' scientific process skills were tested by a Wilcoxon test; the result indicated a positive change from pre-test ($M_{\text{pre}} = 13.2$) to post-test ($M_{\text{post}} = 17.25$) with $z = -3.52$ and $p < 0.001$.

Some investigated experienced RE teachers' perceptions of the possible effects of robotics on students' skills and abilities. For example, Khanlari (2013) interviewed seven RE teachers who, based on their own experiences with teaching robotics, believed that robotics learning could help develop students' creativity, critical thinking, team-working, and collaboration skills, self-confidence, independence, and sociality.

Because the current study focuses on students' problem-solving and computational thinking skills, more examples related to these two skills will be provided and discussed in the later sections (i.e., section 2.4.1 and 2.4.2).

Students' attitude gains in RE. Regarding the attitude aspect, most of the studies concerned students' attitude toward STEM. For example, Nugent and colleagues (2010) integrated robotics and geospatial techniques into a week-long intervention and examined the participants' (i.e., middle school students) STEM learning and attitudes. The results demonstrated that this kind of short-term learning significantly enhanced students' learning interests in STEM-related subjects ($t(123) = 6.92$, $p < .001$, $d = .62$). In AÇIŞLI (2016), students' attitudes toward STEM were tested. The results suggested that students' attitudes, in terms of mathematics ($M_{\text{pre}} = 4.19$, $M_{\text{post}} = 4.54$, $z = -2.57$, $p = .01$), science ($M_{\text{pre}} = 4.27$, $M_{\text{post}} = 4.44$, $z = -2.28$, $p = .02$), and engineering ($M_{\text{pre}} = 4.44$, $M_{\text{post}} = 4.47$, $z = -1.85$, $p = .06$), were significantly enhanced by robotics activities. However, the main limitation of AÇIŞLI's study is that the "T" for technology was not included in the test and no reason was given.

In addition, girls' attitudes and interests were usually focused in RE research. For example, in Weinberg and colleagues' (2007) seven-week robot project, called *Botball*, girls' attitudes toward engineering careers were enhanced because "*Botball* decreas[ed] the acceptance of traditional gender roles" (p. 3).

It is notable that the results from previous studies were not always positive. For example, in Hussain et al.'s (2006) LEGO robotics training program, a trained group's problem-solving ability was found to decrease (from M_{pre} = .70 to M_{post} = .65) and have statistical significance (p = .023 < .05). Because most of the previous studies claimed that RE is effective in promoting students' learning, it is necessary for us to critically think about two issues. The first is whether those studies selectively reported "good" results. This query stems from the fact that some previous studies claimed that they measured a set of learning outcomes yet reported only those with significant results. For example, according to Kazakoff and Bers (2008), students' robotics and programming concepts were assessed by 5-point Likert scales, the levels of motivation and engagement were evaluated by 3-point Likert scales, and sequencing abilities were scored by the researcher with sequencing cards. However, the results section reported only the results of students' enhanced sequencing abilities. For this issue, the current study suggests that whether the result is "good," "bad," or temporarily non-proved, we should faithfully provide it to readers, as every result is valuable for RE research. The second issue is: Why can some students benefit from RE while others cannot? Previous studies, whether or not the result is significant, rarely addressed this issue due to the *technological determinist paradigm*, as presented before, so that individuals' diversities, especially those of individuals who did not gain benefits from robotics learning, would be overlooked. The current study addresses this issue by focusing more on the students' learning process. As stated by Jung and Won (2018), research focusing on students' learning process is still rare, while process-focused studies would be meaningful because "the findings can suggest more responsive and accessible pedagogical implications for teaching young children" (p. 10). By focusing more on students' learning process, the current study could not only discover what students might learn in a robotics summer camp but also explore the kinds of interactions that might effectively develop students' learning outcomes.

2.2.7 Summary

The five themes of relevant previous studies on the RE field were reviewed above. From the general trending of RE research, we can see that RE is a promising area with increasing concerns in recent years. Robotics education, which is the main concept of this study, has been defined. As for the theoretical foundations of RE research, two dominant theories were Piaget's constructivism and Papert's constructionism, while previous studies narrowed down these theories into a technological determinist paradigm. Regarding methodology issues, most of the previous studies were short-term, non-experimental with various data collection methods, and concerning primary school participants. As for the settings, previous studies were located in the informal environment such as summer camps, adopting LEGO robot kits and graphical programming interfaces. In terms of the research focuses, previous studies focused on students' gains in terms of knowledge, skill, and attitude in learning robotics. Both positive and negative results regarding students' robotics learning were found in previous studies.

With discussions of previous studies, there are some implications for the current study. Firstly, when adopting constructivism and constructionism theories, it is important to avoid the technological determinist paradigm, which requires us to pay more attention to how students learn rather than what they learn, and to students' learning process rather than their learning outcomes. Therefore, this study concerns not only students' gains but also their classroom interactions. Secondly, both positive and negative results are valuable, and we should focus on students who cannot gain positive results from robotics learning. In this research, those who can or cannot benefit from robotics learning were selected and compared as focal samples and their performances were tracked day-by-day to characterize effective classroom interactions. Thirdly, a short-term informal learning environment has its advantages in RE research, so that this study was conducted at a short-term robotics summer camp. Last but not least, for experimental research design, a control group is not a necessity in RE. Thus, research involving only one intervention group could also be valuable (more rationales regarding this design will be presented in the methodology).

2.3 Classroom Interactions in RE

Classroom interaction was commonly focused by sociolinguistics scholars with an interest in students' language learning. Since the 1960s, due to growing student diversity in the classroom, increasing areas of concern were teaching, learning, and classroom interactions for purposes of understanding what happened between teachers and students. At that time, research on classroom interactions was mainly observational and quantitative, focusing on the effects of teachers' behaviours on students' learning outcomes (Markee, 2015). With the wide use of distance learning techniques, interactions in e-learning settings also became a heated research focus. Moore (1989) characterized three types of interactions in distance education, labelled as learner-content interaction, learner-instructor interaction, and learner-learner interaction. Several studies focused on classroom interactions in physics education (Kalu & Ali, 2004) and science education (Chin, 2006), though little research explored classroom interactions in the robotics teaching-learning environment because robotics education is a relatively new field. However, interactions in the RE classroom are also significant. Most of the previous studies on RE classroom interactions probed the interactions between teachers and students (Liu, Lin, Liou, Feng, & Hou, 2013) or among peers (Yuen et al., 2014). However, in the current study, conducted in a robotics learning environment, robot (learning kits) could be deemed as a crucial component of interacting with students. As stated by Shin and Kim (2007), younger children tend to interact with robots as their peers. Therefore, the uniqueness of interactions in the RE should be noted.

In the current study, **classroom interactions** refers to *a series of mutually-influenced behaviours that take place between two objects in [a] classroom* (Mattheos, 2004). Specifically, this study explores interactions between students and their teachers, peers, and robots, namely: Student-Teacher interaction, Student-Student interaction, and Student-Robot interaction. Relevant RE studies of K-12 education were reviewed.

2.3.1 Student-Teacher (S-T) Interactions

The interaction between students and teachers is the most classical interaction within the classroom (Kalu & Ali, 2004), which plays an important role in students' learning outcomes. As stated by Englehart (2009), "the way that teachers and students interact is a critical factor in determining student outcomes." (p. 731). In a meta-review conducted by Wang and colleagues (1990), among 30 variables, the positive and productive S-T interaction was identified as one of the most important to good learning outcomes. Therefore, revealing the effective patterns of S-T interactions is important in RE field, as it can provide useful data for RE curriculum development, and for RE teachers to better manage learning environment, especially when RE is a relatively new field (Kucuk & Sisman, 2017).

Existing studies of S-T interactions in RE focused on patterns of S-T interactions, applying the S-T interaction model, and pedagogical design. Regarding patterns of S-T interactions, for example, by coding all Student-Teacher interactions as seven categories (three are about teachers and four are about students), Kucuk and Sisman (2017) attempted to determine the S-T interaction patterns in robotics instruction. Finding and solving problems was coded as one of students' behaviour, while the frequency of this behaviour was counted as the least frequent one (*N* = 216 out of 13,629) among four student behaviours. Therefore, it suggested that "teachers should encourage students to identify problems and produce solutions for improving their problem solving skills" (p. 41). The findings also showed that students usually have the brick assembling (N = 5663) and ideas sharing (N = 2476), while teachers usually provide guidance (N = 3646) and ask questions (N = 1097). Additionally, the researchers found that gamification of the process and collaborative works could be positive factors in Student-Teacher interaction in RE. Kucuk's study revealed details regarding how teachers and students interact with each other in RE. They also pointed out that, to foster effective teaching and learning, it is crucial to explore the interactions between teacher and student in RE. The major limitation of their study was that they focused only on one-to-one robotics instruction, which is not a common form of RE. Nonetheless, they proposed that, in future studies, S-T interactions should be focused in a one-to-many environment to further explore the interaction process. The same research design for probing one-to-one S-T interactions in RE and similar results could also be found in the study of Liu and colleagues (2013). While comparing students' finding and solving of problem behaviours in two studies, Liu's study found that the frequency of this behaviour was ranked in third place rather than the last place in Kucuk's study. However, Liu's study was having only one children participant and one teacher participant, which is the main limitation.

Larkins, Moore, Rubbo, and Covington (2013) attempted to apply the cognitive apprenticeship model, which is an S-T interaction model (Collins, Brown, & Holum, 1991; Järvelä, 1999), at a robotics summer camp for middle school students. Besides LEGO Mindstorms robot kits, engineering notebooks were used for scaffolding students' robotics learning. This study precisely described how the teacher adopted the cognitive apprenticeship model’s six methods (i.e., modelling, coaching, scaffolding, articulation, reflection, and exploration) to facilitate students’ learning process and how students were interacted with and benefited from cognitive apprenticeship methods. After students’ interest in STEM subjects and careers was measured in the pre- and post-tests, the results indicated that students showed more interest after learning than before, although in the pre-test, students’ attitudes were already positive. Unfortunately, no significant difference was found in terms of math and STEM careers. Another finding regarded students’ engineering design skills qualitatively evaluated by students’ engineering notebook images, showing that their articulation of design plan and goals had improved. Larkins’ study showed the successful use of the cognitive apprenticeship teaching model in RE. This study claimed that, “over the course of the two-week period, students learn problem-solving skills, computational thinking, and scientific reasoning using cognitive apprenticeship techniques.” (p. 92) However, it did not evaluate students’ PS and CT nor provide evidence of students’ PS/CT development. Hence, it is hard to convince the readers to agree with this assertion.

Other literature talked about teachers’ pedagogical design in RE, and previous studies documented different attempts to use pedagogical models, such as the Engineering Design Process (Bers et al., 2014; Bers, 2008), Technological Pedagogical Content Knowledge and Technology Mapping models (Ioannou & Angeli, 2016), cooperative learning and problem-based learning (Mosley, Ardito, & Scollins, 2016), project-based learning (Barak & Assal, 2018), and so forth. According to the systematic review done by Xia and Zhong (2018), 64.64% of the papers they reviewed (a total of 22 papers) discussed pedagogical interventions. Nonetheless, most of them were more concerned with the effectiveness or outcomes of adopting a certain pedagogical intervention rather than the process of students' and teachers' interactions.

Among various pedagogies, problem-based learning (PBL) or project-based learning (PjBL) were particularly concerned by this research since these pedagogies are closely related to problem-solving abilities. Relevant literature were examined (i.e., Carbonaro, Rex, & Chambers, 2004; Ford, Dack, & Prejean, 2006; Grandgenett, Ostler, Topp, & Goeman, 2012; Price, Rimington, Chew, & Demidenko, 2010; Zhao, 2018) , however, it is seen that problem-solving ability was not the focus of previous studies adopting PBL/PjBL; instead, they were more interested in students' STEM achievements, attitudes, and other learning outcomes. Moreover, studies on adopting PBL/PjBL in RE emphasized more on the teacher's behaviours, such as how teacher used PBL/PjBL to facilitate students' learning. Different from previous studies, the current research focused on the classroom interactions, and how those interactions facilitate students' PS and CT, which provides more details about how students' learning outcomes were shaped from the perspective of classroom interaction.

2.3.2 Student-Student (S-S) Interactions

Similar to the S-T interaction as mentioned before, in Wang and colleagues' (1990) meta-review, "a peer culture supportive of academic achievement" (p. 37) was also listed as one of the most important factor influencing students' learning outcomes. There are numerous studies showing that facilitating appropriate S-S interactions is helpful for cultivating students' cognitive abilities (see e.g., King, 2002; Messer, Joiner, Loveridge, Light, & Littleton, 1993; Rardin & Moan, 1971). However, in RE field, it is seen that there is no extensive work on this kind of interaction. It is of significance to research peer interaction in RE, in particular which kinds of interaction are effective for developing students' transferrable skills, because related evidences give hints at how to construct a productive learning climate in the robotics classroom.

In 2014, Yuen and colleagues tried to code and count the frequencies of students' interactions within a group. The results showed that students communicated well with each other when they worked as groups, because robotics concepts were still relatively novel to all students. Therefore, students were rarely exclusionary toward each other. However, in building and programming tasks, students did not tend to collaborate with others; instead, they preferred to work individually. The researchers also found that, when working as groups, students had to wait for their turn to use the robot kits or computers. During this waiting time, some students got lost in off-task things while others were able to observe their peers. Yuen's study offered some evidence that robotics activities may naturally lead to a collaborative situation; however, it should be critically considered as to whether forcing students to work in groups is a necessity, as students, especially young-age children, naturally prefer to create their own robots and might get lost when impatiently waiting for their turns to use the robot kits or computers.

Lee, Sullivan, and Bers (2013) examined nineteen kindergarten students' peer-to-peer interactions at a robotics summer camp, in which they had been assigned into two kinds of classes: One provided students with a structured environment while the other gave more freedom to students. The results indicated that students who were allowed to explore robotics freely performed better in interacting with other students. Despite the limited sample size (*N* = 19) and short duration (five days), Lee's research is a good example of probing peer interactions in RE, though they discussed only one type of peer interaction: collaboration. In fact, competition, fighting, joking, and so on are all important types of interactions.

Another RE study concerning peer interactions was that of Jordan and McDaniel Jr. (2014). How students responded to their group members' attempts to manage uncertainty was observed, and both socially supportive and unsupportive peer responses were found. When students received supportive peer responses, they participated in collaborative ideas and problem-solving. When they received unsupportive responses, they might be trapped into *badgers*, withdraw their engagement, modify their approach, or seek outside resources. Jordan's study described an explicit path of how students' problem-solving, especially uncertainty management, was influenced by peer interactions.

2.3.3 Student-Robot (S-R) Interactions

Student-Robot interactions, which have rarely been discussed in previous studies, are also a meaningful part of classroom interactions that is significant to students' learning outcomes and worth studying in RE. In some traditional subjects, such as mathematics and physics, students have few opportunities to directly interact with an object. Differently, students who learn robotics rely on "talking" to the robot (i.e., inputting design ideas to build and program a robot) and "listening" to the robot's *responses* (i.e., observing the robot's outputs and reactions). Afterward, they will decide how to debug and modify their ideas about building or programming the robot. As a result, students learn robotics from such kinds of Student-Robot interactions. That is why Jung and Won (2018) suggested that future RE studies should concentrate more on learners' voices for probing "their dynamic and complicated interactions with robotics kits" (p. 15).

What should be noted is that Student-Robot interaction in the RE field is a different concept from child-robot interactions in the social robot field. Child-robot interactions focus on the interactive behaviours between humanoid social robots (such as tutor robots) and users (i.e., children) in many kinds of learning settings (e.g., language learning, special education needs, etc.) (Belpaeme et al., 2013), while Student-Robot interactions focus mainly on the process between learners and programmable robot kits, in RE settings.

Limited studies have probed Student-Robot interactions in RE. For example, in Levy and Mioduser's (2010) observation, *Participatory Investigations* were found to be one manner in which students bodily interacted with the robot; “in such interactions, the child’s role shifts from designer and observer to that of participant” (p. 28). Nevertheless, Levy’s study did not further probe other possible interactions between students and the robot.

Most of the previous studies focused only on one-way behaviours of students handling of robots. For example, in Yuen et al. (2014), when describing students’ tasks in the robotics project, the researchers determined that there are seven kinds of tasks students should carry out with the robot kits: “building, programming, testing, debugging” (p. 41). Finally, they found that building and observing were the two most favourable parts among students, while programming was students’ least favourite part. Their study described the clear process of how a student made a robot, which was simply a one-way process of *doing*. However, without the information about how students interacted with the robot, such as how they dealt with the unexpected actions of a programmed robot and how they fixed the bugs, we can hardly know why students rarely have interest in the programming part—which is a significant part for the cultivation of future STEM talents.

2.3.4 Summary

Three interactions in the RE classroom were reviewed above. The Student-Teacher interaction is the most classical one in almost all RE environments. The majority of the prior studies probed this kind of interaction from the perspective of pedagogical interventions, though those studies focused mainly on students' outcomes in a given instructional process rather than on how students and teachers communicate and interact with each other.

Meanwhile, some studies focus on the Student-Student interaction in RE. They talked mainly about students' collaborative interactions, while other kinds of interactions were overlooked. The Student-Robot interaction in RE was an under-researched point, while previous literature called for attention to be paid to this interaction (such as Jung & Won, 2018).

Additionally, the significant limitation of previous studies of interactions in RE was that they rarely discussed the relationship between students' classroom interactions and learning outcomes. Therefore, it is hard for us to know which interactions might be effective for developing students' learning outcomes. The current study filled this gap.

2.4 Students' Learning Outcomes in RE

The current study concerns mainly on students' problem-solving (PS) competencies and computational thinking (CT) skill, which are two important learning outcomes investigated by other RE researchers (e.g., Barak & Zadok, 2009; Bers, 2010; Bers et al., 2014, etc.). Robotics learning contains two important aspects: building physical artifacts and programming. Correspondingly, solving problems in both physical building and algorithmic programming is significant. In the current study, PS competency is about students' general designing ideas for solving building and programming problems, while computational thinking skill is more about solving algorithmic problems when programming and debugging.

Problem-solving was usually regarded as the most crucial cognitive activity (Jonassen, 2000). Both educators and educational researchers paid a significant amount of attention to developing students' PS competencies. We can see, from many countries' educational governmental documents for cultivating future students, that PS was identified as a significant skill for all ages of students. In the Programme for International Student Assessment (PISA) sponsored by the Organization for Economic Co-operation and Development (OECD), problem-solving competency was listed as one important domain of students' capacity. In 2015, *collaborative problem-solving* replaced *complex problem-solving* and, for the first time, became one domain evaluating students' ability. In mainland China, problem-solving was included in the framework of *Core Competencies and Values for Chinese Students Development* (Lin, 2017). **Problem-Solving (PS)** in the current study means *the capacity to define, understand, and resolve problem situations when the method of solution is not immediately certain* (Martinez, 1998; OECD, 2013). This research used problem-solving *competencies* rather than problem-solving *skills* because it not only focused on students' PS skills, but also their PS self-efficacy (Maydeu-olivares & D'Zurilla, 1997).

Unlike PS competencies, computational thinking (CT) skill was rarely mentioned by Chinese governmental documents because it is a new concept, usually only mentioned in

Information Technology (IT) education, coding education, and even robotics education. It shared some features of PS skill; however, while PS skill aims to solve various problems, computational thinking is more about "algorithmically solving problems and … the acquisition of technological fluency" (Bers, 2010, p. 3). In other words, CT skill is also about solving problems, but it requires students to apply the proper algorithm and technological knowledge (i.e., programming) to solve computational problems. In the current study, **Computational Thinking (CT)** refers to "*a problem-solving process that includes formulating problems; logical organization of analysis of data; representation of data through abstractions; identifying and automating solutions through algorithmic thinking; analysing and implementing possible solutions; and generalizing and transferring the problem-solving process*" (Leonard et al., 2016, p. 868).

As discussed above, PS competencies and CT skill are both significant learning outcomes in RE, so these two learning outcomes were further examined in the present study. Relevant previous studies concerning these two capacities of students in RE will be critically reviewed in the following sections.

2.4.1 Problem-Solving (PS) in RE

Because robotics learning is concrete and tangible, it offers students opportunities to solve problems with realistic constraints (Petre & Price, 2004). Some RE studies focus on students' problem-solving. For example, Wagner's (1998) pioneering study measured students’ problem-solving skill in three modes of learning environments: traditional science learning (without manipulatives and computers), robotics power by battery (without programming), and robotics powered by computer (with programming), with two different lengths of duration (one week and two week); several meaningful results were found. Firstly, in the one-week treatment, problem-solving measures of the computer robotics group increased from the pre-test (M_{pre} = 50) to the post-test (M_{post} = 51); similar growth was found in the battery robotics group (M_{pre} = 50, M_{post} = 50.3), while no comparison analysis was conducted between the pre- and post-test, and the statistics of the traditional group were not offered. Secondly, in the two-week treatment, students in the computer robotics group performed better on the post-test (M_{post} = 82.7) than on the pre-test (M_{pre} = 68.1), while the battery robotics group showed a similar shift (M_{pre} = 73.2, M_{post} = 83.3). Thirdly, a comparison of computer and battery groups in either a one-week or two-week duration found no significant results. Fourthly, a comparison of all three modes and of battery/computer robotics versus traditional learning, in either a one-week or two-week duration, indicated that students gained more in battery or computer robotics learning than in traditional learning. The results above suggested that engaging students in manipulative experiences (i.e., battery robotics and computer robotics) might effectively enhance students’ problem-solving. Another interesting finding was that, compared to students using only battery robotics, students in the computer group with opportunities to program with robots showed better performance in programming problem-solving. Although Wagner’s study was conducted 20 years ago, it showed a rigorous research design (a true experimental design) with a large-scale sample size (364 in total), which provided rich empirical evidence and critical insights for RE research. In Wagner's time, other pioneering studies'—such as Shimabukuro (1989), Winer (1989), Flake (1990), and so forth—on students' PS in RE were also important. These studies provided some evidence about RE's effectiveness in promoting students' PS. However, "little can be found definitively as to whether robotics improves science and problem-solving test scores" (Wagner, p. 153).

Recently, more studies have concerned students' PS in RE. For example, in an interpretive study conducted by Norton et al. (2007), the activity theory model was used for analysing interactions "between teachers' beliefs and pedagogical practices (teachers' activity) upon middle school students' approaches to designing, problem solving and constructing in a Lego Robotics environment (student activity)" (p. 263). Two important assertions were made. The first is that "teachers' goals and beliefs mediated the formation of different learning communities" (p. 265). The second is, "More holistic problem solving approaches were associated with students' systemic use of flow charting" (p. 268). For the second assertion, which is related to PS in RE, researchers found that some students were willing to use a flow chart because they thought it might assist in their logic expression, while other students thought that this tool was irrelevant and even hindered the problem-solving process. Nevertheless, findings suggested that those who can utilize a flow chart in systemic ways would perform better in problem-solving.

The study of Castledine and Chalmers (2011) observed how students changed their problem-solving strategies in LEGO robotics learning. The findings showed that most of the students (45%) chose to change software programming, 30% changed numerous trial runs, and 20% changed robot design. Also, the researchers proposed that students' efficacy in solving problems was promoted through engaging in robot design/software programming, revising designs in repeated processes, and reflecting on their work. Consequently, their problem-solving and metacognitive skills were improved.

As Martinez (1998) stated: "young children are the most natural problem solvers." However, Atmatzidou and colleagues (2018) pointed out that without appropriate guidance, young students may get lost and the skill of solving problems might not be taught properly. A total of 18 hours of robotics learning involved two groups: one with "strong" guidance by the teacher and the other with "minimal" guidance. The study compared students' opinions about whether robotics learning increased their problem-solving skills. The results of a Mann-Whitney U test suggested that strong guidance was more effective in developing students' problem-solving skills as compared to minimal guidance ($M_{\text{minimal}} = 3.83$, $M_{\text{strong}} = 4.56$, $U = 26.5$, $p = .022$). This study reminded us of the possible influence of Student-Teacher interaction on students' PS skills in RE.

Rather than directly focusing on students' problem-solving skills, Sullivan (2008) attempted to observe how students utilized thinking skills and science process skills to solve robotics problems. The study findings indicated that both the robotics environment and the pedagogical method emphasizing open-ended and extended inquiry may lead to the utilization of thinking skills and science process skills in solving robotics problems.

In the research conducted by Barak and Zadok (2009), the researchers stated that introducing the instructional elements of problem-solving is necessary in a robotics program. Their study found that students usually generate inventive solutions when solving robotics problems. Specifically, the beginning learners (i.e., the first-/second-year students) intuitively used heuristics search methods based on life experiences but might benefit from learning principles or techniques for solving problems in the third-/fourth-year learning. The studies above encouraged teachers' effective engagement in developing students' problem-solving skills.

In Barak's recent study (Barak & Assal, 2018), a *P3 Task Taxonomy* was presented to distinguish three levels of assignments for junior high school students in RE: *Practice* level with closed-ended tasks and known solutions/answers; *Problem-solving* level with open-ended tasks and unknown solutions/different answers; and *Project*s level with challenging tasks and ill-defined problems. It was assumed that some of the students might be able to complete assignments at the first two levels. The results indicated that around 50% of students could complete tasks at the problem-solving level through independent work, while the rest needed the teacher's assistance. This research contributes to the literature on how robotics learning can serve as a platform for promoting students' problem-solving. Moreover, the P3 Task Taxonomy offered a useful model for RE researchers and educators to understand students' achievement levels. However, they failed to describe how students' problem-solving ability was cultivated. Therefore, some important issues should be further addressed. For example: Given different levels of assignment, might students' problem-solving abilities also be different? Why can some students complete assignments at the problem-solving or highest level while others cannot?

As for RE research in the Chinese context, several studies have evaluated students' PS skills in the RE learning environment. For example, Zhao (2018) conducted a case study at a primary-level school in mainland China, exploring a project-based learning robotics curriculum. Based on the results of evaluating students' PS skills by pre-/post- test, the researcher concluded that students' problem-solving skills were enhanced through four-round 32-hour robotics learning ($t(20) = -2.27$, $p = .034$). In the Taiwan region, Tsei (2009) found that intervention-group (with 15-week robotics learning) students' problem-solving skills were significantly improved as compared to those of the control group ($t(20) = -3.68$, $p = .001$). Although these studies showed the enhancement of students' PS skills given the robotics learning environment, they failed to explain how students' problem-solving skills were developed.

Studies with negative findings should also be noticed. For example, Hussain et al.'s (2006) study of 322 middle school students measured PS skill; however, no significant results were found between controlled and trained groups; for the trained group, an opposite shift from pre-test (M_{pre} = .70) to post-test (M_{post} = .65) was found with p = .023. The researchers did not try to explain this situation, though they called for future studies to probe it.

In summary, several previous studies of students' problem-solving skills in the RE environment noticed the importance of teachers' role in moderating students' skill development, while few studies in the Chinese context concerned this role. Nevertheless, how students interact with teachers/peers/robots and how these interactions influence students' problem-solving development requires more attention and discussion. As for the data collection issue, in addition to quantitative methods measuring students' problem-solving skill (i.e., pre-/post-test, such as Hussain et al. (2006) and Wagner (1998)), classroom observation (such as Sullivan, 2008), and semi-structured interview (such as Barak & Assal, 2018) were mainly employed in relevant studies. This reminds us to use mixed methods to probe students' problem-solving.

2.4.2 Computational Thinking (CT) in RE

Inspired by Wing's (2006) call for concerns on CT, many RE studies attempted to probe students' CT in RE settings in recent years. The literature of Lee and colleagures (2011) described what CT looks like in RE. Three domains of CT were presented: abstraction, automation, and analysis. Specifically, in the RE environment, "Abstraction takes place as students design robots to react to a limited set of conditions that may be encountered in the real world." (p. 34). In the abstraction stage, students must think about how to make their robots interact with the outside world and how to abstract the inputs and outputs in their programming. For the automation domain, the computing device (i.e., the intelligent brick of the robot) executes the program written by the student. Then, students carefully analyse the robot's behaviours and reactions to see whether the robot moves as expected. If not, students analyse the bug's location and return to the abstraction stage. Lee's study provided a useful model for understanding students' CT process in robotics learning. However, the automation stage was described, as only robots execute the program, while students' role in this automation was weakened. In fact, the automation stage provided students with opportunities to collect data from the robot's behaviours and, hence, bring those data to the analysis stage. The current study will enrich this model by describing students' role in the automation stage, which will be presented later in the conceptual framework.

Bers (2010) and Bers et al. (2014) introduced a robotics learning program—TangibleK—for engaging kindergarten students in CT. In the study conducted in 2014, to evaluate students' CT, "four key variables were observed and assessed: debugging, correspondence, sequencing, and control flow" (2014, p. 149) for every six lessons (activities). In terms of debugging skill, students' performance in four debugging steps was measured by a 5-point Likert scale. The results indicated that students' debugging skill was in a range from partially to mostly understanding and applying debugging steps with nearly all mean scores above 3 points. However, no significant growth was found across six lessons. For the other three variables, which was labelled as powerful ideas of programming, most of the students achieved the target level (i.e., mostly complete). The studies of the TangibleK program provided critical insights into CT in RE; however, the main limitation was that these studies failed to address a key component of CT—abstraction (Wing, 2006, 2010). Therefore, their framework seems more like programming design ability rather than CT skill. Interestingly, researchers of the TangibleK program also found that, when the study was conducted in a regular school setting, because the professional level of teachers was hard to control, the results might be influenced by many uncertain situations. Meanwhile, in an experimental setting with dedicated teachers, the teaching and researching process is more successful (Bers et al., 2014). That is why, as mentioned previously, much RE research was conducted in a summer camp/lab environment with dedicated teachers (Jung & Won, 2018). This information reminds us about the importance of choosing an appropriate setting and participants (especially the instructor) in future RE studies.

In Eguchi's (2014) survey study of participants in RoboCupJunior (i.e., RCJ, one of the most popular worldwide robot competitions for primary and middle school students), most of the participants (above 60%) agreed that they learned CT when preparing and participating in RCJ. Specifically, among eight dimensions of CT, all informants thought that they had learned debugging (100%), followed by problem-solving (93%), breaking down a big problem into small pieces (79%), logical thinking (79%), analysing skills (79%), creating a step-by-step procedure (71%), critical thinking skill (69%), and prototyping (64%). Eguchi's study provided a CT model with eight dimensions; however, the main limitation was that it failed to demonstrate the literature foundation underpinning this model. Moreover, it did not directly measure students' CT skill; instead, it simply surveyed students' opinions about whether or not they had learned CT, which might lead to some bias problems because those who participated in a world-class competition could be quite confident in their abilities.

Ioannou and Angeli (2016) attempted to measure and compare middle school students' computational thinking in two learning environments: a robotics environment guided by the framework of *Technological Pedagogical Content Knowledge* and the approach of *Technology Mapping* (experimental group) and 3D interactive programming (control group). The results indicated that the CT measure of the experimental group was significantly higher than that of the control group ($M_{\text{experimental}}$ = 69.41, M_{control} = 76.21, p = .05). Nevertheless, when interpreting this result, the researchers pointed out that the difference was due to the design of the instructional intervention, while the variable of the different tools adopted (i.e., robotics and 3D programming) was not discussed. Therefore, it is hard to reach a conclusion about whether the students' gain in CT was due to the tool adopted or the instructional design, or even the two reasons together.

In Leonard and colleagues' (2016) study, students' computational thinking skill was observed by a rubric with seven components: "Formulating problems, abstraction, logical thinking, using algorithms, analysing and implementing solutions, generalizing and problem transfer, and use of pop gaming culture" (p. 875). Students' performances were rated at three levels: emerging (1), moderate (2), or substantial (3). Three environments were included for probing: gaming only, robotics only, and gaming-robotics combined. However, due to limited participants recruited in the robotics only environment, their results and discussions were based mainly on the gaming only and gaming-robotics combined environment, while the robotics only environment was excluded.

In Atmatzidou and Demetriadis's (2016) study with 164 participants from junior high and high vocational schools, CT skill in RE was measured in five dimensions: abstraction, generalization, algorithm, modularity, and decomposition. Their study was conducted in an after-school environment, including a total of 11 robotics learning sessions (two hours per session). Two intermediate questionnaires were assigned after sessions 4 and 10 (i.e., in the paper, two questionnaires were labelled as CT-4 and CT-10, which could be deemed as pre-test and post-test), and after training, a "think-aloud" tool was used to measure students' CT by their free expression of the thinking process (i.e., labelled as CT-TA in the paper). In a comparison of CT-10 to CT-4, significantly positive shifts ($p < .001$) were found from $M_{\text{ct4}} = 2.69$ to $M_{\text{ct10}} = 2.91$, with $t(163) = -5.27$, while for junior high school students, the shift was nonsignificant ($M_{\text{ct4}} = 2.96$, $M_{\text{ct10}} = 3.08$, $t(88) = -1.91$, $p = .059$). In comparing the junior high school group with the high vocational group in CT-TA, no significant difference was found in four of five dimensions of CT, except for *generalization*. In comparing different genders, "Girls appear in most cases to need more time (training sessions) in order to reach the same skill level as boys" (p. 668). This study offered a meaningful model for exploring students' CT skill, and evidence found in this study allowed future RE research and practice to think more about the difference between boys and girls. Additionally, it recognized the impact of students' attitudes toward different modalities of skill assessment; for example: "When students are asked to provide written evidence of their skills, they might appear to underperform because of poorly following the instructions" (p. 668).

In the study of Chen and colleagues (2017), the CT skill of 37 primary school students was measured in pre-/post-tests. Two different contexts were included in the assessment: everyday scenarios and robotics programming. In a comparison of these two contexts, the results showed that a student gains less in everyday scenarios than in robotics programming. For the robotics-programming context, researchers prepared two forms of programming language: a text-based form (similar to a professional programming language) and a drag-drop form (a graphic language, similar to Scratch). No significant result was found in a comparison of students' CT using these two forms. The CT framework they adopted was the SDARE model, which refers to five CT components: syntax, data, algorithms, representing, and efficient and effective. The overall pre-post results indicated a significant increase in the CT in two observed classes (Class 1: *t* = -3.14, *p* = .002; Class 2: *t* = -3.87, *p* = .001). However, through students' pencil-paper test, researchers found that students had trouble following a given syntax, which might be because some of them used graphic language which does not contain obvious syntax. This finding could inspire future RE practice to demonstrate syntax to students in teaching graphic language. The main limitation of Chen's study was that how students' CT skill was developed is unclear; therefore, more information about students' learning process, especially those parts that might contribute to students' CT development, should be further probed.

In the Chinese context, some empirical studies on students' CT in RE were conducted. For example, in the study of Li, Yang, and Chen (2019), 26 primary school RE teachers from Wuhan city in China were involved in in-depth interviews about how to develop students' CT in RE. Four stages of teaching CT for primary school students were demonstrated: stimulating motivation, pattern construction, implementing creative ideas, and communication and reflection. Their study presented a precise model for understanding the teaching process in RE but failed to explain how students' CT was cultivated in each stage. Some other papers contributed to the Chinese-context research on students' CT, though most of them were more like reflective reports based on the authors' own teaching experiences, without a scientific research design.

Although the relevant studies mentioned above did not directly point out the role of classroom interactions in students' computational thinking, we can see that computational thinking is closely related to students' programming and debugging processes where Student-Robot interaction might occur, as through programming and debugging the robot, students were allowed to interact with their robots. Further attention should be paid to the relationship between classroom interactions and CT skill development.

Regarding the research method of previous studies on CT in RE, few adopted only a quantitative method to measure this kind of skill (such as in Atmatzidou & Demetriadis, 2016). They focused more on the process of which students use computational thinking to solve problems, as computational thinking skill is a multiple-level concept rather than a simple indicator.

2.4.3 Summary

Related literature on students' PS and CT in RE were examined, among which some conducted experimental research comparing the robotics environment to the traditional one (i.e., Chen et al., 2017; Hussain et al., 2006; Tsei, 2009; Zhao, 2018) or comparing students' outcomes before and after learning robotics (i.e., Hussain et al., 2006) for the purpose of investigating whether students could learn PS or CT from RE. Whereas, these studies might fall into a technological determinism (Jung & Won, 2018), as discussed previously. In addtion to robotic technology itself, some other possible factors influencing students' PS and CT development were identified by previous studies, such as the use of different learning tools (Leonard et al., 2016; Norton et al., 2007; Wagner, 1998), the intensity of guidance (Atmatzidou et al., 2018), and pedagogical methods (Ioannou & Angeli, 2016; Sullivan, 2008). Since the research of RE is still in the beginning phrase, it is seen that discussions related to the possible factors influencing students' PS and CT were still immethodical. Some other factors, such as student motivation, self-autonomy, hands-on tasks, and others, which might be significant to students' PS and CT development, were under-researched and calling for future studies. Due to this situation, in the current stage of research, exploring the various possible factors is necessary. As discussed in the section 2.3, classroom interactions are crucial factor determining students' learning outcomes, therefore, this research focused on this factor, although it acknowledges that other factors might have more effects.

2.5 Summary and Research Gaps

Appendix A summarizes RE studies, reviewed above, on classroom interactions and two learning outcomes (i.e., PS and CT). The first three columns contain basic information about reviewed studies, including authors, years, titles, participants, and methods. The fourth column indicates whether the study concerned classroom interactions and whether relevant details were briefly provided. The fifth column indicates whether the study concerned PS/CT; relevant details are also briefly described. Because the main results of those studies were provided in detail in previous writing, the table presents only brief introductions to the results. From the reviews and the summary table, the author noticed some gaps in previous studies. Firstly, among RE studies on classroom interactions, S-T interaction was the most mentioned one, while most of the studies focused only on the teaching process/pedagogical issues, so students' engagement was under-researched; S-R interaction was the least mentioned one in previous studies, which is the main issue calling for more attention (Jung & Won, 2018). Secondly, abundant evidence of students' gains in PS and CT were found in previous RE studies, while few of them were conducted in the Chinese context. Thirdly, although some studies concerned both classroom interactions and learning outcomes, the relationship between classroom interactions and learning outcomes was less discussed. Nevertheless, previous RE studies provided critical insights into the possibility of probing in students' classroom interactions, PS, CT, and relationships among them. The current study also gains benefits from those studies in terms of conceptual framework and research design. Studies that contribute to the conceptual framework of the current study will be discussed in the following sections.

2.6 Conceptual Framework

Concerning the research gaps discussed before, the current study, underpinned by constructivism and constructionism theories, aims to probe three interactions in the RE classroom: Student-Teacher interactions, Student-Student interactions, and Student-Robot interactions. Additionally, students' learning outcomes (i.e., PS and CT) will be examined. Moreover, the relationship between interactions and learning outcomes will be discovered. The conceptual framework can be seen in Figure 2. Theoretical foundations and concepts of this framework will be discussed.

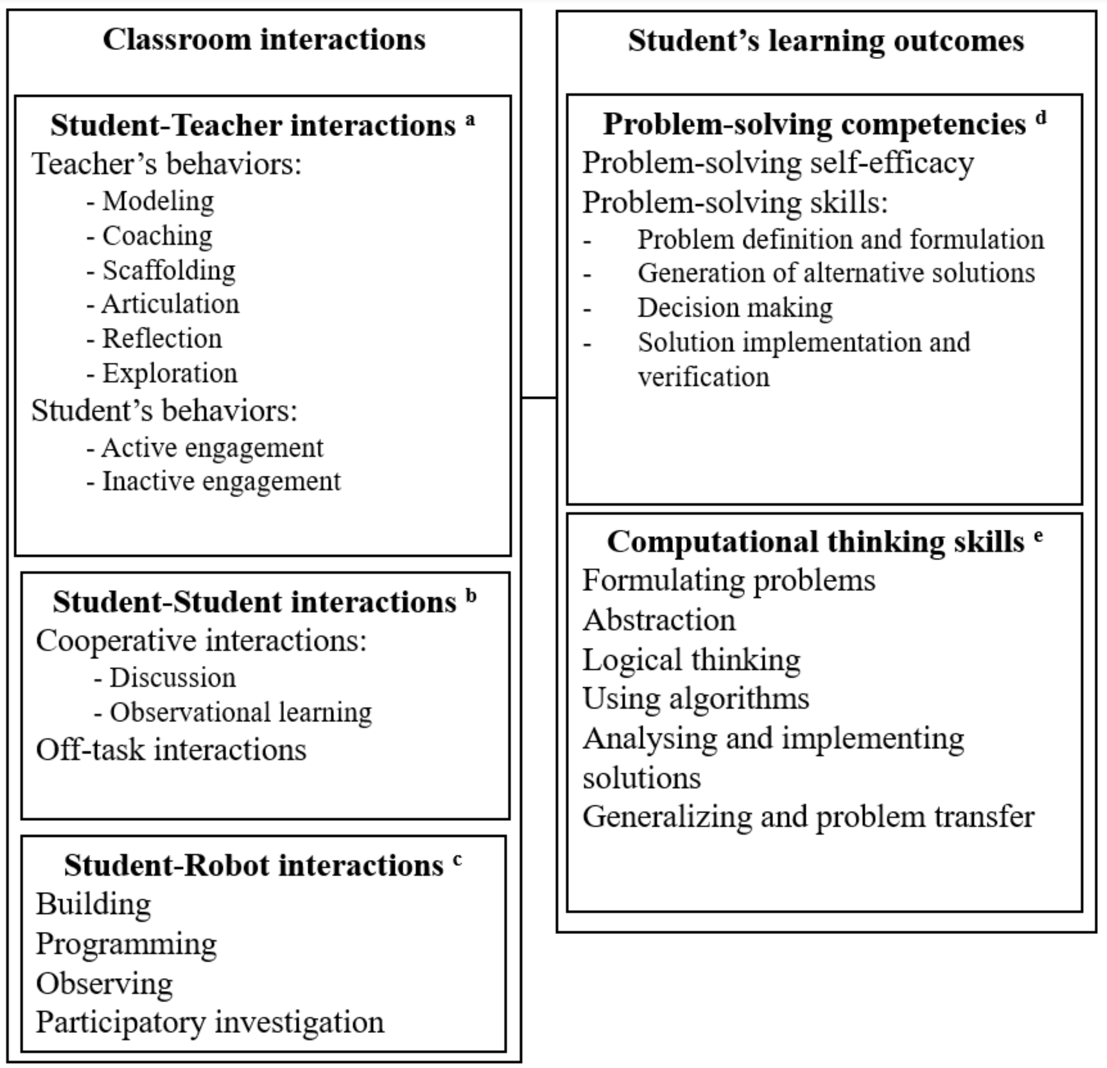
2.6.1 Theoretical Foundations

As reviewed previously, in the RE field, Piaget's constructivism and Papert's constructionism were two dominant theories (Anwar et al., 2019; Eguchi, 2012; Jung & Won, 2018; Mubin, Stevens, Shahid, Mahmud, & Dong, 2013). The current study is underpinned by both theories. Piaget’s constructivism regards students’ learning as constructing the experience of interacting with their environments (Eguchi, 2012; Jung & Won, 2018), which inspires this study to focus on how students learn by interacting with teachers, peers, and robots. Also, constructivism acknowledges that children need an object to put their hands on and think with (Ackermann, 2001; Eguchi, 2012). Thus, this study demonstrates how students’ thinking was shaped by hands-on activities with the robot. Papert’s constructionism theory is well-fitted to the RE area, as it asserts that learning occurs when children are “engaged in designing and building their own personally meaningful artifacts and sharing them with others in a community” (Bers, 2008, p. 125) which focuses on *learning by design* (Bers, 2010); meanwhile, robots are naturally tangible and must be physically manipulated as part of the learning process (Mubin et al., 2013). Therefore, this study allows students to design and create their own robots and probes in students’ gains in such hands-on experience. To address the problem of the technological determinist paradigm (Jung & Won, 2018), this study conducted process-oriented research, not only measuring students’ gains before and after learning but also focusing on students’ learning process. Therefore, this study do not try to explain students’ learning outcomes by the use of technologies, but discovers effective classroom interactions that might contribute to the development of students’ PS and CT.

2.6.2 Classroom Interactions

As discussed previously, in the current study, **classroom interactions** refers to *a series of mutually-influenced behaviours that take place between two objects in [a] classroom* (Mattheos, 2004). First of all, an interaction is not a single behaviour, but a series of behaviours. Secondly, these behaviours can be verbal or physical. Thirdly, the influence between two objects should be mutual. Fourthly, an interaction involves two objects. The core of this definition is "mutually-influenced". Sometimes, singly way expression could be "mutually-influenced". For example, when the teacher was modelling, and students were carefully listening. It was a singly way expression, but, when the teacher was influencing students, he was also being influenced by students' performances, facial expressions, or other behaviours. However, if a student looks away from the teacher while the teacher is lecturing, such behaviour will not be counted as interaction.

Specifically, this study focuses on three kinds of classroom interactions: S-T, S-S, and S-R interactions, which are all related to the student. These student-related classroom interactions will be further explicated.


Classroom interactions
Student-Teacher interactions a
Teacher's behaviors:
- Modeling
- Coaching
- Scaffolding
- Articulation
- Reflection
- Exploration
Student's behaviors:
- Active engagement
- Inactive engagement
Student-Student interactions b
Cooperative interactions:
- Discussion
- Observational learning
Off-task interactions
Student-Robot interactions c
Building
Programming
Observing
Participatory investigation
Student's learning outcomes
Problem-solving competencies d
Problem-solving self-efficacy
Problem-solving skills:
- Problem definition and formulation
- Generation of alternative solutions
- Decision making
- Solution implementation and verification
Computational thinking skills e
Formulating problems
Abstraction
Logical thinking
Using algorithms
Analysing and implementing solutions
Generalizing and problem transfer

Note.
[a] Adapted from Collins et al. (1991), Järvelä (1999), and Larkins et al. (2013).
[b] Adapted from Azmitia (1988), Duckworth (2005), and Yuen et al. (2014).
[c] Adapted from Levy & Mioduser (2010) and Yuen et al. (2014)
[d] From D'Zurilla's series works with associates (e.g., Chang, D'Zurilla, & Sanna, 2004; D'Zurilla & Maydeu-Olivares, 1995; Maydeu-olivares & D'Zurilla, 1997).
[c] From Atmatzidou & Demetriadis (2016), ISTE (2014), and Leonard et al. (2016).

Figure 2. Conceptual Framework of the Current Study.

Student-Teacher interactions. As reviewed previously, several studies concerned S-T interactions in RE (Atmatzidou et al., 2018; Ioannou & Angeli, 2016; Kucuk & Sisman, 2017; Larkins et al., 2013; Lee et al., 2013; Li et al., 2019; Liu et al., 2013; Mosley et al., 2016; Norton et al., 2007; Yuen et al., 2014). Among these studies, some (see, e.g., Kucuk & Sisman, 2017; Liu et al., 2013) presented a model of the patterns of both teachers' and students' behaviours in S-T interactions, containing three behaviours of the teacher (i.e., providing guidance, correcting mistakes, and asking questions) and four behaviours of the student (i.e., assembling bricks, playing with the products, sharing and expressing, and finding and solving problems). The advantage of this model is that it properly generalizes the patterns of students' and teachers' behaviours in the RE setting, which is especially helpful for understanding what happens in the RE classroom. However, the weakness of this model is that it is too general; thus, it is hard for us to build a relationship between S-T interactions and students' learning outcomes and to know which kinds of interactions might be effective in the development of PS or CT. For example, one of the teacher's behaviours—providing guidance—could not explicitly describe the teacher's methods adopted for guiding students' learning and thinking. Another considerable problem is that students' behaviours contained in this model, such as assembling bricks and playing with the products, are more like kinds of S-R interactions. Also, sharing and expressing is more closed to S-S interactions. Due to the considerations above, this study did not use this model when analysing S-T interactions.

Some other studies concerning S-T interactions attempted to apply pedagogical models in RE settings (e.g., Barak & Assal, 2018; Bers, 2008; Ioannou & Angeli, 2016; Lee et al., 2013). Among these models, the Engineering Design Process (EDP) was among the most cited ones (Bers et al., 2014; Bers, 2008; Eguchi, 2012). It generally contains six stages: ask (identifying problems), imagine, plan, create, test & improve (redesign), and share (communicate) (Bers et al., 2014; Eguchi, 2012). This model described an effective process for students to solve problems like an engineer; however, it is more like a problem-solving process, while the manner in which the teacher can successfully engage students in the stages of this process should be further discussed.

Consequently, this research decided to adopt the cognitive apprenticeship model, consisting of six teaching methods (Collins et al., 1991) to guide the teacher's teaching behaviours and to explore students' corresponding learning behaviours. The six teaching methods are: modelling, coaching, scaffolding, articulation, reflection, and exploration. This model is selected for analysing S-T interactions for several reasons. Firstly, this model showed potential in cultivating students' PS and CT, as these methods help students think visibly (Collins et al., 1991; Järvelä, 1999). Secondly, this model was applied in the RE setting, showing potential in effectively teaching PS and CT (Larkins et al., 2013). Lastly, previous studies suggested using this model to explore S-T interactions rather than concentrating only on teaching behaviours (Collins et al., 1991; Järvelä, 1999), which is well-fitted to the purpose of this study.

In this model of S-T interaction, six crucial concepts are modelling, coaching, scaffolding, articulation, reflection, and exploration. Next, definitions of these concepts, as well as teachers' roles and students' active engagement in each method, will be interpreted.

Modelling "involves an expert's performing a task so that the students can observe and build a conceptual model of the processes that are required to accomplish it" (Collins et al., 1991, p. 13). In modelling, the teacher's role is to demonstrate robotic techniques of building and programming that can be used to complete the lesson task (Larkins et al., 2013). Correspondingly, students' active engagement in modelling involves carefully observing the teacher's performance and building their own conceptual model.

Coaching "consists of observing students while they carry out a task and offering hints, scaffolding, feedback, modelling, reminders, and new tasks aimed at bringing their performance closer to expert performance" (Collins et al., 1991, p. 14). Therefore, the teacher's role in coaching is to observe students' working process and support students in working through difficulties (Larkins et al., 2013). Students' active engagement involves identifying and discussing the problems with the teacher and following the teacher's coaching.

Scaffolding "refers to the supports the teacher provides to help the student carry out the task" (Collins et al., 1991, p. 14). Other than suggestions or direct help, the teacher could use physical support to scaffold students (Collins et al., 1991). In RE, the teacher could utilize resources such as robot kits, flow charts, and computers for performance scaffolding (Larkins et al., 2013). Thereby, students' active engagement is to follow suggestions and utilize scaffolds provided by the teacher.

Articulation "involves any method of getting students to articulate their knowledge, reasoning, or problem-solving processes" (Collins et al., 1991, p. 14). The teacher's role is to encourage the student to articulate and refine their thinking process; thereby, students' active engagement includes thinking aloud following the teacher's facilitation.

Reflection "involves enabling students to compare their own problem-solving processes with those of an expert, another student, and ultimately, an internal cognitive model of expertise" (Collins et al., 1991, p. 14). Therefore, the teacher should lead students and provide an environment in which they can critically reflect on their own work. This requires students' active engagement in communications with the teacher and peers.

Exploration "involves pushing students into a mode of problem solving on their own. Forcing them to do exploration is critical, if they are to learn how to frame questions or problems that are interesting and that they can solve" (Collins et al., 1991, p. 14). Hence, the teacher should not only prescribe tasks but also provide less-structured and challenging tasks for students to freely experiment with their own innovative ideas (Larkins et al., 2013). Students' active engagement in exploration, therefore, means to explore the challenging tasks provided by the teacher and to gain more experience in problem-solving and computational thinking.

As discussed above, the conceptual framework includes both a teacher's cognitive apprenticeship teaching and students' engagement in each teaching method. Additionally, students' engagement could be further divided into active and inactive engagement, as we should acknowledge that some students are unwilling to cooperate with the teacher's teaching. Consequently, six concepts of the cognitive apprenticeship model, along with students' active and inactive engagement, are included in the conceptual framework.

Student-Student interactions. Limited studies have explored Student-Student interaction in the robotics learning environment (Jordan & McDaniel Jr, 2014; Lee et al., 2013; Yuen et al., 2014). Among these studies, Lee's study concentrated on the frequency of S-S interactions; it did not provide a model to describe the patterns of S-S interactions. According to Yuen et al. (2014), cooperative and competitive interactions are two typical patterns of peer interaction in robotics learning. However, the current study did not involve competition activities at the summer camp, due to considerations such as the limited class time and young children's abilities to complete the lesson tasks; therefore, competitive interactions were not expected in the current study. In Hussain et al. (2006), one way by which students learn is to cooperate with others, though only seeking help was pointed out; this could not properly describe how students' different thoughts interacted.

For cooperative S-S interaction, the current study attempts to further break it down into two concepts: discussion and observational learning. In the study of Jordan and McDaniel Jr. (2014), students' responses to peers could be classified into supportive response and insupportive response, which could also be labelled as *ideas sharing and conflicting* (Azmitia, 1988). To better generalize students' ideas sharing and conflicting, in the conceptual framework of the current study, the concept of *discussion* is used to echo the constructivism theory because “it was always Piaget’s belief that discussion among peers was the most likely means of moving learners beyond their current understanding” (Duckworth, 2005, p. 262). Additionally, as found in the RE study of Rogers and Portsmore (2004), most student learning stems from discussion. Furthermore, observational learning is included in the framework of S-S interaction because it might be effective for developing students’ PS when “novels” can observe and learn from “experts” (Azmitia, 1988).

Discussion herein refers to students’ in-class sharing and the conflicts between their ideas. In peer discussion, students freely share their thoughts, designs, thinking processes, PS and CT strategies with each other, or conflicts with others’ ideas. The key point of peer discussion is students’ idea exchange, so that different ideas could be swapped (Duckworth, 2005), leading to the “collective creation of knowledge” (Eguchi, 2012, p. 21).

Observational learning involves students’ roles as experts and novels. In a class, some students who perform better than others might play experts’ roles; novels can observe the experts’ PS and CT process and learn from them. In observational learning, peer discussions might also occur.

In addition to the cooperative S-S interactions discussed above, we should also notice that off-task interactions, such as unstructured chatting, playing with the robot or peers, and other interactions not related to the lesson task are also part of S-S interactions (Yuen et al., 2014), which should be included in the conceptual framework.

Student-Robot interactions. As reviewed previously, Student-Robot interaction was rarely discussed in RE studies. However, the significance of this kind of interaction in RE should get more attention (Jung & Won, 2018). To begin understanding the patterns of Student-Robot interaction, we should first know how student handle robots. As proposed by Wagner (1998), robotics involves two areas of education: manipulatives and computerized language. In other words, in robotics learning, students build manipulatives and programs with computer language for the manipulatives. Therefore, two patterns of students' hands-on behaviours regarding the robot could be conceptualized: building and programming. In Yuen et al. (2014), four typical patterns of students' hands-on behaviours were found, including: building, programming, testing, and debugging. Testing and debugging could also be deemed as parts of the programming process. Moreover, students' building and programming are all based on the robot's responses. During the process of "talking" (i.e., building and programming) to the robot and "listening" (i.e., observing) to the robot's responses (i.e., behaviours and reactions), S-R interactions occur. Therefore, building, programming, and observing are included in the conceptual framework. Additionally, as suggested by Levy and Mioduser (2010), students' bodily interactions with the robot, namely, *participatory investigation*, is also an important part of S-R interactions.

Specifically, in *building* the interaction process, students need to not only build their own robot but also repeatedly refine the structure of the robot according to the robot's responses. In the *programming* process, students write programs, test whether the programs could properly lead the robot to solve problems, and debug the program. In the *observing* process, the student observes the robot's behaviours and reactions to collect information which might help with the building and programming process. As for *participatory investigation*, "the child's role shifts from designer and observer to that of participant" (Levy & Mioduser, 2010, p. 28), and the student bodily interacts with the robot to investigate its behaviours and reactions.

2.6.3 Students' Learning Outcomes

The current study focused on two potential learning outcomes in RE: *problem-solving competencies* and *computational thinking skills*. In selecting suitable models for observing and assessing PS competencies and CT skills, two rationales were followed. First, the competencies/skills included in the model should be observable and measurable, as one of the purposes of this study is to evaluate students' learning outcomes and see the potential changes before and after the robotics summer camp. Therefore, the model with a corresponding measurement tool is preferred. Second, the model should be well-underpinned by relevant theories, such as problem-solving and computational thinking theories. Next, selected models and concepts included are discussed.

Problem-solving competencies. In the current study, **Problem-Solving (PS)** means *the capacity to define, understand, and resolve problem situations when the method of solution is not immediately certain* (Martinez, 1998; OECD, 2013). As reviewed previously, there were many studies concerning students' PS in RE; however, some claimed that students learned PS but did not evaluate students' PS (Larkins et al., 2013), some focused on PS process and strategies rather than on PS competencies/skills (Barak & Assal, 2018; Barak & Zadok, 2009; Castledine & Chalmers, 2011; Jordan & McDaniel Jr, 2014; Norton et al., 2007), and some measured students' PS but failed to provide an explicit model of PS (Atmatzidou et al., 2018; Hussain et al., 2006; Tsei, 2009; Wagner, 1998; Zhao, 2018). Consequently, this study turned beyond the RE field to find a suitable PS model.

A frequently-cited model for measuring problem-solving competencies was Heppner and Petersen's (1982) Problem-Solving Inventory (PSI), containing three concepts: problem-solving confidence, approach-avoidance style, and personal control. Although this model, as well as the tool, were widely adopted by numerous studies (Heppner, Witty, & Dixon, 2004), as critiqued by D'Zurilla and Maydeu-Olivares (1995), its main weakness is the lack of a clear link to problem-solving theories. In fact, the generation of PSI's items was originally based on D'Zurilla and Goldfried's problem-solving theories in 1971, but in the factor exploration process, a new model with a three-factor solution was suggested by statistics; consequently, the final model could not be properly linked to the original theoretical model. Also, the original theoretical model was modified by D'Zurilla and colleagues (D'Zurilla, 1986; D’Zurilla & Nezu, 1990) into two domains: Problem-Solving Self-Efficacy (PSSE) and Problem-Solving Skills (PSS). The corresponding tool for measuring these PS competencies was validated by Maydeu-Olivares and D’Zurilla (1997), who refined and shortened Heppner’s 32-item PSI to a 16-item scale. Because the shorter PSI (SPSI) was developed in a more scientific way, and the underpinned theoretical model is well-matched to the tool (Maydeu-Olivares & D’Zurilla, 1997), compared to Hepper’s PSI, the current study finally chose the SPSI and the underpinned theoretical model to observe and evaluate students’ PS competencies.

In D’Zurilla’s PS model, problem-solving self-efficacy refers to “belie[f] in one’s personal ability to solve problems successfully” (Chang et al., 2004, p. 15). In other words, a better problem solver should be more optimistic and confident when solving problems.

Problem-Solving Skills refers to four skills: problem definition and formulation, generation of alternative solutions, decision making, and solution implementation and verification (Chang et al., 2004; D’Zurilla & Goldfried, 1971). Those who can better perform these skills were labelled as *constructive problem solvers*; on the other hand, those who failed to perform these skills were deemed as *dysfunctional problem solvers* (Chang et al., 2004).

In *problem definition and formulation*, “the problem solver tries to clarify and understand the problem by gathering as many specific and concrete facts about the problem as possible, identifying demands and obstacles, and setting realistic problem-solving goals” (Chang et al., 2004, p. 16). A *problem*, herein, refers to a “situation or task (present or anticipated) that demands a response for adaptive functioning but no effective response is immediately apparent or available to the person or people confronted with the situation because of the presence of one or more obstacles” (p. 12). Therefore, properly identifying a problem could be deemed as the key sign of the students’ gain of this skill.

In the *generation of alternative solutions,* “the person focuses on the problem-solving goals and tries to identify as many potential solutions as possible, including both conventional and original solutions.” (Chang et al., 2004, p. 16). Hence, this skill concentrates on students’ ability to find solutions to the problem. It expects a better problem solver to generate more alternative solutions, regardless of whether or not the solution is inventive.

About *decision making*, “the problem solver anticipates the consequences of the different solutions, judges and compares them, and then chooses the ‘best’ or potentially most effective solution” (Chang et al., 2004, p. 16). This skill requires students to simulate the possible outcomes of generated alternative solutions and to make decisions about which solution is the most effective one by critically judging and comparing.

As for *solution implementation and verification*, the major sign of gaining this skill is that the student carefully manages and evaluates the outcome of the chosen solution after implementing it (Chang et al., 2004, p. 16). A better problem solver should have the ability to carry out the chosen solution and to verify whether the actual outcome is as expected.

To sum up, the current study adopts D'Zurilla's PS model and the SPSI to evaluate students' PS competencies, because of some rationales and considerations, as mentioned previously. The conceptual framework includes two domains of PS: problem-solving self-efficacy and problem-solving skills. For problem-solving skills, there are four major skills, problem definition and formulation, generation of alternative solutions, decision making, and solution implementation and verification.

Computational thinking skills. As reviewed before, there are numerous studies concerning students' CT in RE at the K-12 level (Atmatzidou & Demetriadis, 2016; Bers, 2010; Bers et al., 2014; Chen et al., 2017; Eguchi, 2014; Ioannou & Angeli, 2016; Lee et al., 2011; Leonard et al., 2016; Li et al., 2019). When selecting a suitable CT model for the current study, two rationales—the same as for selecting the PS model—were considered. Among all relevant literature reviewed previously, Li et al. (2019) were firstly excluded because they focused only on teaching strategies and did not provide a CT model for evaluating students' CT. Other studies providing CT models are summarized in Table 1. Lee et al.'s (2011) CT model was well underpinned by Wing's original CT theories (Wing, 2006, 2010), but it is more like a CT process model rather than a skill model. The models of Bers (2010), Bers et al. (2014), Eguchi (2014), Ioannou & Angeli (2016) , and Chen et al. (2017) lack the main element of CT, *Abstraction*, which was deemed as the essence of CT in the CT theories (Wing, 2006, 2010). Consequently, the models of Atmatzidou and Demetriadis

(2016) and Leonard et al. (2016) were focused on. These two models shared some elements, such as abstraction, generalization, and algorithm. Meanwhile, *Modularity* in Atmatzidou's model could be included in *Generalizing and problem transfer* in Leonard's model, as modularity is to generalize the often-used commands that could be transferred to solving the same or different problems (Atmatzidou & Demetriadis, 2016). Moreover, *Decomposition* in Atmatzidou's model could be merged into *Formulating problems* in Leonard's model, as decomposing problems into smaller ones is part of properly formulating problems.

Table 1

CT Models Adopted by Previous RE Studies

Studies	CT models
The chosen model	
International Society for Technology in Education (2014)	Formulating problems, Abstraction, Logical thinking, Using algorithms, Analysing and implementing solutions, Generalizing and problem transfer
Other models	
Lee et al. (2011)	Abstraction, Automation, and Analysis
Bers (2010)	Debugging, Correspondence, Sequencing, and Control flow
Bers et al. (2014)	
Eguchi (2014)	Problem Solving, Debugging, Prototyping, Decomposition, Logical thinking, Creating step-by-step procedure, Analysing Skills, Critical Thinking, Iteration, and Debugging
Ioannou & Angeli (2016)	Verbal description, Pseudocode, Logical diagram, Sequential structure, Branch structure, Designate an algorithm, and Steps for the development of a program
Atmatzidou & Demetriadis (2016)	Abstraction, Generalization, Algorithm, Modularity, and Decomposition
Leonard et al. (2016)	Formulating problems, Abstraction, Logical thinking, Using algorithms, Analysing and implementing solutions,

	Generalizing and problem transfer, and Use of pop gaming culture
Chen et al. (2017)	Syntax, Data, Algorithms, Representing, and Efficient and effective

Leonard's model is preferred, as discussed above, and provided a valid and reliable tool for measuring students' CT skills (i.e., Computational Thinking Rubric, *CTR*, in Leonard et al., 2016, pp. 874-875). However, the element of *Use of pop gaming culture* included in the model is not suitable for the current study. The original model adopted by Leonard et al. (2016) is from the International Society for Technology in Education (ISTE) (2014), and the use of pop gaming culture was not included in the ISTE's original CT model. Therefore, the current study follows the ISTE's model and removes the use of pop gaming culture from the CTR. Also, **Computational Thinking (CT)** was defined as "*a problem-solving process that includes formulating problems; logical organization of analysis of data; representation of data through abstractions; identifying and automating solutions through algorithmic thinking; analysing and implementing possible solutions; and generalizing and transferring the problem-solving process*" (Leonard et al., 2016, p. 868). The current study adopts this definition because it clearly describes the characteristics of the computational thinking skill, which is aligned with CT theories (Wing, 2006, 2008, 2010). Specifically, six key concepts are included in the conceptual framework: Formulating problems, Abstraction, Logical thinking, Using algorithms, Analysing and implementing solutions, and Generalizing and problem transfer.

Formulating problems requires students to identify problems and formulate them "in a way that enables us to use a computer and other tools to help solve them" (ISTE, 2014, p. 1). This process involves students' ability to break down general problems into smaller ones (Atmatzidou & Demetriadis, 2016). The main goal of formulating problems is to make problems solvable.

Abstraction herein means "representing data through abstractions such as models and simulations" (ISTE, 2014, p. 1). When designing robots, the student simulates how the robot might react to given conditions by using abstract thinking (Lee et al., 2011). The main sign of gaining this skill is that the student uses programming concepts to describe concrete situations.

Logical thinking means students' skill in "logically organizing and analysing data" (ISTE, 2014, p. 1). Those who think logically might organize their programs in logical ways, and they might use terms like "because…so…" or "if…then…" to express their thoughts.

Using algorithms evaluates students' ability to use proper commands following the basic nature of algorithms (ISTE, 2014). The key sign of gaining this skill is students' use of proper programming logics (i.e., conditional logic, iterative logic, or parallel logic) in ordering their robots to complete the task.

Analysing and implementing solutions is when the student automates solutions and evaluates the effectiveness of them (ISTE, 2014). This skill could also be deemed as debugging skills (Leonard et al., 2016), which is for finding the faulty parts (i.e., Bugs) in the program.

The last skill, *generalizing and problem transfer*, is to generalize and transfer the "problem solving process to a wide variety of problems" (ISTE, 2014, p. 1). A better computational thinker can encapsulate the problem-solving process (such as modularity) and expand solutions to cover the same or different problems, rather than solve certain problems in a one-off manner. The main sign of gaining this skill is when the student solves different problems through solutions used before.

From the above, the ISTE's (2014) model and the adapted Leonard's CTR are adopted in this study for observing and evaluating students' CT skills. Six CT skills have been conceptualized.

2.6.4 The Relationship Between Classroom Interactions and Students' Learning Outcomes

The current study focused mainly on understanding the relationship between three classroom interactions (i.e., S-T, S-S, and S-R interactions) and two learning outcomes (i.e., PS and CT) in the RE setting. Therefore, the conceptual framework shows a line between classroom interactions and students' learning outcomes. This relationship contains two dimensions. The first is the relationship between students' time spent on classroom interactions and their learning outcome shifts. The second is, the potential influences of effective classroom interactions on students' learning outcomes.

Chapter 3: Methodology

In this chapter, the research purposes and questions will firstly be proposed. Then, the mixed-method research approach adopted by this research will be presented and justified. After that, background information about the robotics summer camp will be described. Additionally, the participants, instruments, and procedures of data collection will be introduced. Moreover, the manners in which the qualitative and quantitative data were analysed will be discussed, respectively. Also, the procedures and results of the pilot study for validating a PS instrument will be presented. Lastly, the ethical issues and the limitations of methodology will be discussed.

3.1 Research Purposes and Questions

This study had three purposes. Firstly, this study was to explore the features of students' PS and CT in robotics learning. Secondly, it was to evaluate students' PS competencies and CT skills in robotics learning, especially the possible changes in these two learning outcomes. Thirdly, it was to identify the relationship between classroom interactions and students' PS competencies and CT skills.

Accordingly, this study was guided by three research questions:

Research Question 1: What are the features of students' problem-solving and computational thinking in a robotics learning environment?

This research question was for understanding what students' problem-solving and computational thinking look like in robotics practice. It focused on three aspects. The first is the kind(s) of problems that students usually encountered in the robotics learning environment, which might require students' PS competencies/CT skills to solve. The second is the process of problem-solving/computational thinking that students usually followed. The third is the relationship between PS competencies and CT skills. Other features emerged from the data were also embraced.

Research Question 2: What are the changes in students' problem-solving competencies and computational thinking skills in a robotics learning environment?

This research question was for evaluating students' PS competencies and CT skills at the robotics summer camp of this study. According to the previously proposed conceptual framework, problem-solving self-efficacy, four problem-solving skills, and six computational thinking skills were focused on.

Research Question 3: What are the relationships between three classroom interactions (i.e., S-T, S-S, and S-R interactions) and students' learning outcome development (i.e., PS competencies and CT skills)?

This research question concerned two dimensions of classroom interactions: students' time spent on classroom interactions and the effective kinds of classroom interactions. For the time spent on classroom interactions, this study counted students' time spent on S-T, S-S, and S-R and explored its correlation to the change in students' learning outcomes found in RQ2.

For the effective kinds of classroom interactions, among several kinds of interactions proposed in the theoretical framework, this study attempted to explore the kinds of interactions that might be effective for the development of students' PS competencies and CT skills.

3.2 Mixed-Methods Research

The current research adopts a mixed-method approach, which is "an approach to inquiry involving collecting both quantitative and qualitative data, integrating the two forms of data, and using distinct designs that may involve philosophical assumptions and theoretical frameworks" (Creswell, 2014, p. 41). A research approach involves three components: philosophical worldview, design, and research methods (Creswell, 2014), which means "in planning a study, researchers need to think through the philosophical worldview assumptions that they bring to the study, the research design that is related to this worldview, and the specific methods or procedures of research that translate the approach into practice" (p. 43). Accordingly, this mixed-method research holds a pragmatic philosophical worldview, employs qualitative and mixed-method designs, and adopts mixed methods.

3.2.1 Pragmatic Philosophical Worldview

Pragmatism, as a research paradigm stemming from Dewey's philosophy agenda, was commonly associated with mixed-method research in social research (Creswell, 2014;

Feilzer, 2010; Morgan, 2014). It regards the phenomenon of the real world as multi-level, so it falls to "quantitative methods to measure some aspects of the phenomenon in question and qualitative methods for others" (Feilzer, 2010, p. 9). Therefore, pragmatism attempts to break down the dualists of post-positivists and constructivists (Feilzer, 2010; Morgan, 2014). In other words, when "post-positivists claim that the world exists apart from our understanding of it, while constructivists insist that the world is created by our conceptions of it" (Morgan, 2014, p. 1048), pragmatism believes that "either the nature of the outside world or the world of our conceptions are just discussions about two sides of the same coin" (p. 1048). Also, pragmatism aims to solve problems "and then us[e] pluralistic approaches to derive knowledge about the problem" (Creswell, 2014, p. 50), so that "the pragmatist researcher is able to select the research design and the methodology that are most appropriate to address the research question" (Kaushik & Walsh, 2019, p. 261).

According to the discussions above, the current research adopted the pragmatic research paradigm to shape the methodology because: Firstly, this study did not focus on only one aspect of the phenomenon; for example, concerning classroom interactions, both time spent on classroom interactions and the patterns of different kinds of interactions were valued, so the pragmatic paradigm allowed the investigator to collect information from different aspects. Secondly, although this study was basically guided by some conceptions of the world, it was also open to exploring uncertainties of the world (Feilzer, 2010); for example, when exploring classroom interactions in RE, this study was not limited to a concern about those kinds of interactions listed in the conceptual framework, which means that any other interactions emerging from the data were embraced. Thirdly, to provide the best understanding of problems, this mixed-method research adopted multiple methods and different forms of data collection and analysis (Creswell, 2014), including pre-/post-test, rubric scoring, semi-structured interview, classroom observation, and both quantitative and qualitative analysis methods.

3.2.2 Research Designs

Because this study held a pragmatic worldview in terms of the methodology, it embraced any designs that would be effective for answering the research questions. Specifically, the qualitative design was mainly adopted to address RQ1 (i.e., the features of PS/CT), as this RQ aim to explore and describe "the lived experiences of individuals about a phenomenon" (Creswell, 2014, p. 54), such as characterizing students' PS/CT process by their experiences. In addition to the qualitative method, document analysis or content analysis on the previous robotics learning activities have also been carefully considered. However, it is hard to find the documents of previous robotics learning activities, so there were little contents about students' PS and CT could be used to analyse. Moreover, some features of PS and CT can only be found in classroom observation and interview, such as the process of PS and CT.

For RQ2 (i.e., the changes of PS/CT), the convergent parallel mixed-method design was used. Convergent parallel design is "a form of mixed-methods design in which the researcher converges or merges quantitative and qualitative data in order to provide a comprehensive analysis of the research problem" (Creswell, 2014, p. 55). This study, for example, used both quantitative data (i.e., the results of the pre-/post-test and rubric scoring) and qualitative data (i.e., interviews and classroom observation) as triangulation to comprehend students' change in PS competencies and CT skills in RE.

For RQ3 (i.e., the relationship between classroom interactions and PS/CT), similarly, the convergent parallel mixed-method design was adopted. For example, in this study, both quantitative results (i.e., correlation analysis between time spent on classroom interactions and the shifts of PS/CT) and qualitative findings (i.e., identifying which classroom interactions were effective for the development of PS/CT from interviews and observation) were used as triangulation to address the RQ3.

As discussed above, the pragmatic research paradigm guided this study to adopt different designs for addressing particular research question(s). Therefore, this study believes that various kinds of research designs are not absolutely isolated from each other. For example, when "identifying which classroom interactions were effective for the development of PS/CT from interviews and observation" mentioned before, an explanatory sequential design was followed (Creswell, 2014), as this study firstly examined students' PS/CT performance and then explained students' changes (both positive and negative) through their classroom interactions from qualitative data. Finally, this explanatory finding was mixed with quantitative results (i.e., correlation analysis between time spent on classroom interactions and the shifts of PS/CT), which followed a convergent parallel design.

3.2.3 Mixed Methods

In this study, both qualitative and quantitative data were collected, analysed, and interpreted, so that "the researcher makes inferences across both the quantitative and qualitative databases" (Creswell, 2014, p. 57). Specifically, quantitative methods such as pre-/post-test and rubric scoring (close-ended), and qualitative methods such as semi-structured interview and classroom observation (open-ended), were employed in this study, aiming to provide a comprehensive understanding of students' learning in RE. The details about data collection and analysis will be presented later.

It is important to justify the use of mixed methods (Creswell, 2014). For this research, the use of both quantitative and qualitative methods was suitable because, by connecting, merging, or/and embedding both forms of data, research purposes and questions could be better addressed. Quantitative data allow us to understand the general situation of students' classroom interactions (i.e., time spent on each interaction) and learning outcomes (i.e., the measures of students' PS and CT). By conducting quantitative analysis, we can easily determine the correlation between (time spent on) classroom interactions and PS/CT performance, as well as the change in students' learning outcomes before and after training.

Nevertheless, in relying only on numeric data, the significant information buried in students' actual learning process might be overlooked. Therefore, qualitative data play a more important role in this study, as they enrich the quantitative conclusions and providing a comprehensive understanding of the features of PS/CT, classroom interactions, learning outcomes, and the relationships in between. Additionally, as suggested by Cohen, Manion, and Morrison (2011), involving "the use of more than one method in the pursuit of a given objective" (p. 197), *triangulation between methods* occurs. As shown in the Table 2, multiple methods were employed to answer each research question. By connecting, merging, or embedding both forms of data, different aspects of information could be triangulated to each other.

Table 2

Research Methods Adopted by This Research

RQs	Research methods			
	Pre-/post test	*Rubric scoring*	*Interview*	*Observation*
RQ1			√	√
RQ2	√		√	√
RQ3	√	√	√	√

To conclude, the current study is mixed-method research, holding a pragmatic philosophical worldview, employing qualitative and mixed-method designs and adopting both qualitative and quantitative methods to collect, analyse, and interpret data. Although this study was posited as mixed-method research, it was more concerned with the qualitative parts of the data as compared to the quantitative data. It is because this research is process-oriented, which focuses more attention on students' learning process rather than the measurement of students' learning outcomes. In other words, quantitative data collected in this study provide a general skeleton of situations, while the qualitative data play a more important role, as muscles to construct holistic knowledge.

3.3 The Robotics Summer Camp

The main intervention of this research was the robotics summer camp where participants were all involved. Changes in students' learning outcomes during the summer camp were measured. Since the focus of this research is not on evaluating this summer camp, so no control group was set up. This section will introduce the robotics summer camp in terms of its setting, schedule, the robot kits, the programming interface, curriculum design, and instruction methods. These components of the summer camp introduced herein served as background information that allows the readers to discover more details about this research. Moreover,

3.3.1 Setting

This study was conducted at a robotics summer camp at The Children's Palace of Liuzhou City (a southern city of mainland China). The robotics summer camp is an extracurricular activity that has been widely adopted by previous RE studies (see, e.g., Larkins et al., 2013; Lee et al., 2013; Yuen et al., 2014). As reviewed previously, RE is still widely deemed “an extra-curricular activity and a part of informal education” (Mubin et al., 2013, p. 5), especially at the K-12 level. This might be because informal learning sessions, such as after-school workshops and summer camps, have advantages such as a short-term duration and minimum curriculum design and teacher training (Mubin et al., 2013). Also, researchers of the TangibleK program (Bers, 2010; Bers et al., 2014) found that, when the study was conducted in a school regular setting, because the professional level of teachers was hard to control, the results might be influenced by many uncertainties. On the other hand, in an experimental setting with dedicated teachers, such as at a robotics summer camp, the teaching and researching process will be more successful (Bers et al., 2014). Additionally, robotics activities usually require an environment with specific hardware allowing children to use robot kits and to program with computers, while a formal classroom environment could not provide such supports, especially in the general context of mainland China (Zhang & Zhang, 2008). Due to the considerations above, this study took place at a robotics summer camp.

The choice of the summer camp was based on three considerations: First, the summer camp is located in the centre of the city, and it is not so far from every districts of the city, so the students willing to come to the camp were diverse. Second, this summer camp was free of charge, so students from various socioeconomic backgrounds could be involved. Also, the feature of free is important because a paid summer camp may affect students’ performance

— If students know that their parents paid for their summer-camp learning, they may be more active and productive. Additionally, since the researcher hopes the results of this study could be extended to the public-school settings, a free summer camp is ideal. Furthermore, those summer camps that require a fee might concern about their commercial secrets, so they were not willing to let an outsider to record the class activities. Third, this summer camp was provided for students who have no prior robotics learning experiences, so the effect of prior experiences could be easily controlled in such a research concerning on students' changes in learning outcomes. I had also considered some free summer camps on primary school campus, but those on-campus summer camps were for students participating in the competition, and their learning levels were various. Due to reasons above, this summer camp was chosen.

Liuzhou has a mature environment for robotics learning activities. Liuzhou is an industrial city that has paid a significant amount of attention to RE for adolescents. It was in the earliest batch of cities promoting RE. Therefore, robotics learning activities have gained lots of supports from students, parents, and schools in Liuzhou.

The Children's Palaces (in Chinese: 少年宫) are government-funded non-profit recreation centres for children's out-of-school activities in mainland China. Liuzhou's Children's Palace has provided various summer camps, such as robotics learning, dancing, and painting, for all children in Liuzhou during every summer vacation. Participants recruited at the summer camp were from various primary schools in Liuzhou. Because all costs for the summer camps were paid by the local government, participants in the summer camp only need to simply pay the refundable deposit (around $20) for using the facilities (such as robot kits and laptops). For most families, this deposit is affordable.

One multi-media classroom located in The Children's Palace was used for the robotics summer camp in this research. The classroom had two big workbenches, one blackboard, and several chairs. In total, 10 sets of robotics kits and 10 laptops were prepared for robotics teaching.

3.3.2 Schedule

The summer camp lasted for four weeks from 8th July to 3rd August and included a total of twelve lessons (three times per week). Each lesson lasted for 90 minutes. In total, there were 18 learning hours. Due to the limitation of the facilities, one class can accommodate only 10 students at most. Therefore, all 40 students were assigned to four different time periods. Class A was held during the morning section and Class B was held during the afternoon section, every Monday, Wednesday, and Friday. Class C was held during the morning section and Class D was held during the afternoon section, every Tuesday, Thursday, and Saturday. All classes were the same in terms of the teacher, venue, curriculum, facilities, duration, and so forth; the only difference was the time period. Also, data from different classes were not compared; instead, all participants from the four classes were regarded as one group when conducting data analysis.

3.3.3 The Robot Kit: KAZI EV5

KAZI EV5 is a programable robot kit developed by KAZI Robotics Education Corporation (Shenzhen, China). The significant trait of KAZI EV5 is that the function and quality are almost the same as those of the Mindstorms Education EV3 set from LEGO, but the price is much lower (i.e., around $280 cheaper than LEGO). For most families in China, the KAZI robot kit is affordable. The KAZI EV5 contains one programmable brick (Intelligent Brick), two large motors, one medium motor, two touch sensors, one colour sensor, one ultrasonic sensor, one gyro sensor, one USB cable, and 822 other elements (such as connectors, beams, wheels, etc.). Generally, one set of KAZI EV5 is for at most two students, though in this study, each student has their own robot kit.

3.3.4 The Programming Interface: Scratch

In the summer camp, Scratch, a visual programming language developed by the Massachusetts Institute of Technology (MIT), was provided for students to write programs and download programs to the Intelligent Brick. As a visual programming language, Scratch is popular in the RE field, especially for young children, as “Scratch was designed with a low floor, or the ability to create very simple programs with very little formal training, but also with high ceilings, of the ability to create very sophisticated applications” (Olabe, Olabe, Basogain, & Castaño, 2010, p. 359). Hence, compared to text-based languages used by professional programmers (such as Java, C, Python, etc.), Scratch is more accessible to young students. With Scratch, children can easily program by dragging together graphical and colourful blocks to implement their designs (Bers, 2010; Maloney et al., 2010).

This study used a revised version of Scratch 2.0. The revised Scratch is for KAZI’s Intelligent Brick to execute the commands. The main functions of the revised Scratch are the same as those of the original version, though the revised Scratch defined a new block called “electronic module” for controlling motors, sensors, and the Intelligent Brick.

3.3.5 Curriculum Design

This research used a well-developed robotics curriculum which had been implemented for over 10 years. This curriculum was initially designed by RE experts from some authorities (e.g., The Science and Technology Association, China Next Generation Education Foundation) and was further refined through several rounds of implementation with primary school students. The main purpose of the curriculum was to equip students with basic knowledge about building and programming the robot by exploring various tasks. In total, twelve tasks were provided. The curriculum started with an introductory lesson so that students could get access to the robot kits and programming interface. This was followed by lessons of increasing difficulty. By exploring tasks at different levels, students acquired robotics knowledge such as motors, sensors, and programming logics. The details of every lesson’s purposes, content, and assessments (lesson tasks) can be seen in Appendix I.

3.3.6 Instruction Methods

As discussed previously, the summer camp used cognitive apprenticeship teaching methods, including: modelling, coaching, scaffolding, articulation, reflection, and exploration (Collins et al., 1991; Järvelä, 1999; Larkins et al., 2013). Each method was explicitly conceptualized in the conceptual framework. Before the summer camp, the researcher and the teacher discussed, and reached an agreement on, how to properly implement these methods. During the summer camp, every class started with the teacher's modelling. Then students started to explore the lesson tasks, supported by the teacher's coaching, scaffolding, and articulation. When students completed the basic level of the lesson task, they were allowed to explore challenging tasks based on their own interests. At the end of the class, students were required to reflect on their work.

No grouped or paired activity was involved in the summer camp. This is because, during a discussion with the teacher, he mentioned some possible weaknesses in group work, based on his teaching experiences. These included: children prefer to create their own robots, students have to wait for turns to use robot kits and the laptop, and the tasks are simple enough for the students to work on individually. The researcher carefully considered the teacher's suggestions and agreed with him. First, this research did not focus on students' collaboration ability in groups. Second, because the robotics learning environment is naturally open to S-S interactions, forcing students to group together is not necessary. Third, as stated before, in the study of Yuen et al. (2014), students tended to work individually although they were assigned to groups, and when waiting for turns to use the robot kits and computers, some students might get lost in off-task activities. Therefore, this research decided to allow students to work individually, though S-S cooperative interactions were encouraged.

3.4 Data Collection

This section will introduce the participants in this study, the instruments, and the data collection procedures.

3.4.1 Participants

This study recruited 40 primary school students from ages seven to nine (i.e., Grades 1-3) in July of 2019. All students voluntarily register for the summer camp under the accompany and supervision of their parents. The participants' age and prior experience were controlled (i.e., age 7 to 9 and without any robotics or programming learning experiences). No further selection process for those who have been successfully registered. Subsequently, all participants were assigned to four classes based on their preferences, 10 in each, though all classes were the same in terms of the teacher, venue, curriculum, facilities, duration, and so forth; the only difference was the time period. Among the40 students, two of them dropped out during the summer camp, while six of them were absent for at least one lesson.

Consequently, a total of 32 students—10 girls and 22 boys—completed all the lessons and data collection procedures of the summer camp. The background information about those 32 students can be seen in Table 3.

Table 3

Students' Demographic Information

Demographic information	*N* = 32
Gender	
- Male	22(68.8%)
- Female	10(31.3%)
Age	
- Seven-year-old (Grade 1)	6(18.8%)
- Eight-year-old (Grade 2)	16(50%)
- Nine-year-old (Grade 3)	10(31.3%)

In addition to the student participants, one teacher participant was involved. The dedicated robotics teacher was responsible for the teaching tasks of the summer camp. The teacher was a 25-year-old local male with a Bachelor of Education degree. He had three years of experience teaching robotics to primary school students, in particular by using the KAZI robot kits and Scratch. It was his second time teaching the robotics summer camp in The Children's Palace. During the initial contact and communication before the summer camp, the teacher showed his passion for teaching robotics and his strong willingness to cooperate with the researcher. Also, his ideas about teaching robotics were impressive and his observation of students' learning was critical; therefore, the author believed that he could provide significant insights and information. According to the comments from his leader, the teacher was proficient in both building and programming robots, experienced in teaching robotics, and kind with students, which made him popular among students and parents. Because of his personal traits, as discussed above, he was recruited to serve as the robotics teacher of the summer camp.

3.4.2 Instruments

This section will introduce four instruments adopted by this research in detail. Firstly, in the pre-post test, Shorter Problem-Solving Inventory (SPSI) was adopted for measuring students' problem-solving competencies. Secondly, in the rubric scoring, Computational Thinking Rubric (CTR) was adopted for scoring students' computational thinking skills. Thirdly, the self-developed interview protocols were used for guiding the semi-structured interviews. Lastly, the self-developed observational form was used for guiding the classroom observation and for the research to make field notes.

Shorter Problem-Solving Inventory (SPSI). As discussed previously, to measure students' problem-solving skills, Maydeu-Olivares and Zurilla's (1997) SPSI (see Appendix B), adapted from Heppner and Petersen's (1982) PSI, was translated and adapted in this study. The original PSI is a widely used self-reported scale for measuring "the person's perception or appraisal of his or her problem-solving capabilities" (Maydeu-Olivares & D'Zurilla, 1997). It contains 32 items with a 6-point Likert-type scale (1 = *Strongly agree* to

6 = *Strongly disagree*). As Heppner and Petersen suggested in 1982, PSI has a three-factor structure: problem-solving confidence, approach avoidance style, and personal control. However, D'Zurilla and Maydeu-Olivares (1995) pointed out that there is a lack of any clear link of the scales of PSI to problem-solving theories. Consequently, it is difficult for researchers to interpret the results collected by PSI. To address this issue, Maydeu-Olivares and Zurilla's (1997) analysis suggested a two-factor structure for only 16 items of PSI containing the problem-solving self-efficacy (PSSE) subscale and the problem-solving skills (PSS) subscale. This shorter PSI (SPSI, see Appendix B) was translated and piloted in this study because: a) the two-factor model is more suitable for interpreting students' problem-solving competencies, as it has strong connections to problem-solving theories; and b) the shorter version might be more suitable for the partcipants of this study who were too young to read and respond to a long scale. After the pilot study, the two-factor model was confirmed and 14 items remained. Of the 14 items, seven are for measuring the students' PSSE, while the other seven are for PSS. Because the finalized version of SPSI used in this research was in Chinese, the Chinese version of SPSI with 14 items was named as CSPSI (see the Appendix B). The pilot study validating the CSPSI will be presented later. Both students and the teacher were required to complete the CSPSI in the pre- and post- tests, so there was one difference between the versions provided to students and to the teacher: "I" in the student version was replaced with "this student" in the teacher version. In the main study, results showed moderate to good reliability of two subscales of CSPSI in both the pre-test and post-

test, with Cronbach’s alpha from .73 to .96 (see Table 4). Namely, data from both students and the teacher showed acceptable reliability coefficients.

Table 4

CSPSI’s Reliability Analysis Results in the Main Study (N = 32)

Cronbach’s alpha	Pre-test		Post-test	
	PSSE	PSS	PSSE	PSS
Students’ perceptions	.955	.964	.930	.728
The teacher’s perception	.831	.931	.822	.937

Computational Thinking Rubric (CTR). As shown in Appendix D, the computational thinking rubric designed by Leonard et al. (2016) was adapted for collecting data on students’ computational thinking skills. The original version of CTR contains seven components of computational thinking: formulating problems, abstraction, logical thinking, using algorithms, analysing and implementing solutions, generalizing and problem transfer, and use of pop gaming culture. This study deleted one CT component, “Use of pop gaming culture,” because that component was beyond the definition of computational thinking by ISTE (2014) and was not related to the purpose of this study. Each component was rated according to one of three levels: emerging (1), moderate (2), substantive (3).

The original version of CTR was in English. Therefore, it was translated into the Chinese version by the researcher and a professional translator. Because the teacher could read English, both the Chinese and English versions were provided to the teacher before the summer camp. Then, the researcher, the teacher, and one expert in the RE field discussed the criteria and examples of different levels of CT skills based on both Leonard's and ISTE's definition and examples for each CT skill. Through this process, the content validity of CTR was ensured.

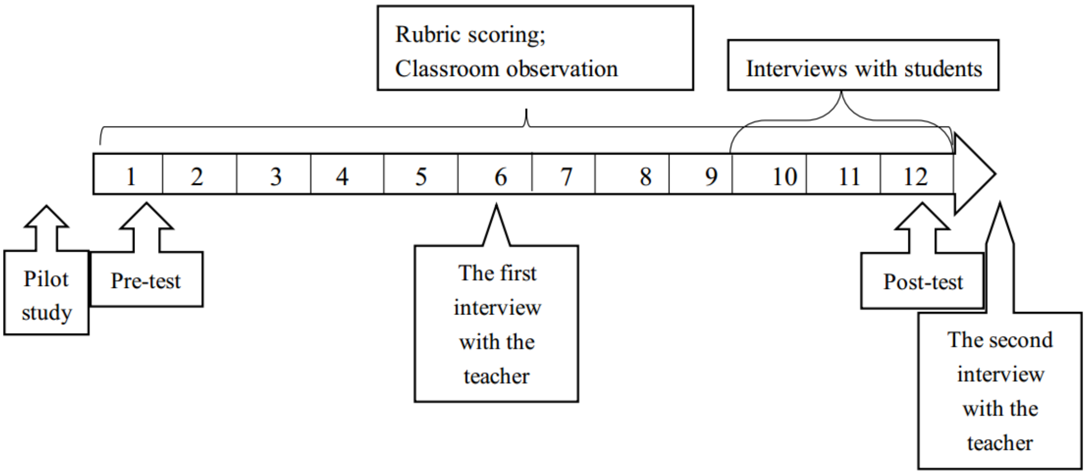
Two raters (i.e., the researcher and the teacher) practiced rating works according to students' performances when they were programming and debugging. As suggested by J. Cohen (1960), inter-rater reliability was measured by Cohen's Kappa test. Each domain of the rubric was tested, respectively. Kappa coefficients of six domains suggested a high level of agreement between two raters ($.54 < \kappa < .92$, $p < .001$).

Interview protocols. The interview protocols for the teacher and students can be seen in Appendix F and G, respectively. The development of interview protocols was initially based on the research purposes and conceptual framework. These protocols contain six parts. The first part is about personal information. The second to fourth parts are about classroom interactions, and the fifth and sixth parts are about learning outcomes. Because the teacher was interviewed twice, the protocol used in the first interview was adapted based on the teacher's responses and suggestions. Therefore, the first interview was like a pilot study; the adapted protocol was then used in the second interview. For the students' interview protocol, the interview was read through by the teacher and, according to his comments, some questions were adapted. The interviews were semi-structured, so protocols were used for scaffolding the researcher to get access to in-depth conversations with the participants. During the interviews, many questions based on participants' responses went beyond the prescribed questions.

Observation form. For purposes of taking field notes during the classroom observation, an observation form was created to help the researcher gather critical information during the classes (see Appendix H). In the observation form, there are places for marking down the details of classroom interactions, such as time of occurrence, participants, and behaviours. Researchers could also write down thoughts about the relationship between these classroom interactions and learning outcomes. The field notes might help the researcher recall critical incidents in the classroom, especially when reviewing the videotapes.

3.4.3 Procedure

Through the methods of pre-/post-test, rubric scoring, classroom observation, and semi-structured interview, both qualitative and quantitative data were collected. The data collection procedure is summarized in Figure 3, and the details are presented as follows.



Note.
The number 1 to 12 refer to Lesson/Day 1 to 12.

Figure 3. Procedure of Data Collection.

Measuring students' problem-solving competencies. Before the summer camp, a pilot study with 118 students was conducted to validate the CSPSI. The procedures and results of the pilot study will be presented later.

As for the main study of this research, on the first and last classes of the summer camp, students were required to complete the CSPSI, which took around 15 minutes. Because the participants were too young to comprehend the items, the researcher explained the meanings of the items one by one to ensure that students' responses matched their own experiences. Additionally, the teacher was asked to score every individual student's problem-solving ability by the CSPSI according to his own observation of students' performance, during the first and last classes. In the pre-test, the teacher's rating was based on each student's performance in the first class; and, in the post-test, the rating was based on student's performance in the last class but not past classes. Because each classroom contained only 10 or fewer students, the teacher's evaluation workload was not too heavy.

Measuring students' computational thinking skill. Before the summer camp, the researcher and the teacher discussed the criteria for scoring. During the summer camp, both the researcher and the teacher scored every students' CT skills in every class, based on their classroom programming and debugging performances and their Scratch programs. Two raters scored students' performances individually to avoid mutual influences.

Classroom observation. The researcher attended every class of the summer camp and took field notes according to the observation form mentioned before. In addition, every class was videotaped by five cameras, as "comprehensive audio-visual recording can overcome the partialness of the observer's view of a single event and can overcome the tendency toward only recording the frequently occurring events" (Cohen et al., 2011, p. 470). Cohen and colleagues (2011) stated that setting either a fixed or movable camera for purposes of classroom videotaping might create the problem of selective data. A second camera (or more) might overcome this, though it is costly, especially in terms of time spent reviewing and analysing the recordings. To record the whole picture of the classroom, four cameras were fixed at the front, back, and two sides of the classroom. Additionally, the researcher held a movable camera to record details about the students' and teacher's behaviours or words. In addition, four audio recorders were set on two workbenches to record participants' verbal interactions more clearly.

Semi-structured interviews. All participants were invited to the interviews and all interviews were recorded. Students were interviewed in groups of two or three. Therefore, 12 groups were interviewed. Each interview lasted for longer than 30 minutes. All interviews were conducted during the last week, when students could reflect on their learning experiences.

The teacher was interviewed twice. The first interview took place on the day of lesson 6, which was the midway point of the summer camp. The second interview was held after the completion of the summer camp.

3.5 Data Analysis

Both quantitative and qualitative methods were used in this study to analyse the collected data. How the data were analysed will be discussed as follows.

3.5.1 Quantitative Analysis

Before the selection of suitable statistical tests, normality tests were conducted because parametric statistics, such as T-Test, should follow the assumption that populations of raw scores form normal distribution (Gravetter & Wallnau, 2014). However, for each paired data to be tested, by observing frequency distribution histograms and graphs of Q-Q plots, this assumption could not be satisfied. Additionally, skewness values, kurtosis values, and z-scores for either skewness or kurtosis were calculated. As suggested by Kim (2013), when the sample size $N < 50$, "if absolute z-scores for either skewness or kurtosis are larger than 1.96, which corresponds with an alpha level 0.05, then reject the null hypothesis and conclude the distribution of the sample is non-normal" (p. 53). However, in most of the cases of both problem-solving and computational thinking measures, because the z-scores were not between -1.96 and 1.96, it was concluded that the distribution of data was non-normal. Moreover, the raw data were hard to transfer to normal distribution by applying mathematical functions, as the data formed a negative skewed distribution or contained lots of decimals. Considering the situations above, non-parametric statistics, such as the Wilcoxon signed-rank test and Spearman's correlation analysis, were more suitable for this research than parametric statistics.

To analyse the quantitative data collected by CSPSI (both students' perception and teacher's perception), the scores of participants' pre-tests and post-tests were compared by the Wilcoxon test in SPSS 21. Because students' perceptions were subjective ratings while the teacher's perceptions were objective (it herein means "external") ratings, scores from two sides should be compared (Schneider, Mädler, & Lang, 2019). Therefore, Spearman's correlation analysis was run to determine the relationship between students' self-reported rating and the teacher's external rating. Conducting correlation analysis was enough to determine the basic relationship between two sets of data (Prince et al., 2008), especially when it is not the main focus of this study. The two sets of data were positively correlated in most aspects, while students' rating was significantly higher than the teacher's rating. It was difficult to know whether students overestimate their own ability or teachers underestimate them, so, to balance the bias, two sets of data were merged by simply counting their mean, and the merged data were tested by the Wilcoxon test again. Also, since there was a positive correlation between the two groups of data, the merged data could also well represent the general situation of students' change in PS. Moreover, merging two sets of data into one can make the subsequent analysis easier.

To analyse the quantitative data collected by CTR (both the teacher and the researcher's evaluation), data from the first six days (i.e., the first two weeks) were compared to data from the last six days (i.e., the last two weeks) by the mean of each six days. Inter-rater agreement was measured by using Cohen's Kappa test. Because the two raters provided third-person perspectives and the results showed strong agreement between the two raters, two sets of data were merged by simply counting their mean, and the merged data were tested by the Wilcoxon test. Additionally, the trending of students' computational thinking skills from the first lesson to the last lesson was analysed. Moreover, the relationship between problem-solving competencies and CT skill was tested by Spearman correlation analysis.

During quantitative analysis of the videotaped/radio-taped classroom, basically based on the conceptual framework and definitions as presented in the section 2.6.2, how much time did each student spend on each type of interaction was measured. When counting the time, the individual was regarded as the unit of analysis, namely every single participants' time spent on interactions were counted. In the process of reviewing the videos, the researcher continuously inquires two questions: Is this student interacting with someone/something? If so, what is the object of his/her interaction (i.e., The teacher, peers, or robots)? When a possible interaction was noticed and the type of such an interaction was initially confirmed, the starting and ending time of this interaction was marked and labelled. Since the researcher could only focus on one or two students at a time, the videos were reviewed for several times, until nearly all possible interactions were found and marked. All marked interactions were carefully checked once again for confirming whether they were consistent with the definition of classroom interaction, whether the type of each interaction was correctly judged, and whether they were aligned with the conceptual framework. When an interaction was confirmed, the duration of this interaction was counted and added to the dataset of its "owner" (i.e., the involved student[s]). As a result, how much time did each student spend on each type of interaction was calculated and used for the next stage of analysis (such as calculations of percentage and correlation).

During the above analysing process, some details should be noted. Firstly, this research counted the duration of time in interactions rather than the frequencies of occurrence, because as defined previously, classroom interaction is not a single behaviour but a series of behaviours. Comparing to counting the frequencies, measuring the time spent on interactions was more suitable for analysing such kind of serial behaviours. Secondly, minutes are units of time when counting, because based on the researcher's in-class observation experiences, most of these interactions were lasting for several minutes. For a few cases that the time of interaction was shorter than one minute, it was counted as one minute. Such a counting method does not have a significant impact on the reliability of results, because these were only a small number of cases, and they occurred about the same number of times in each interaction. In addition, such accuracy is sufficient for calculating correlations. Thirdly, to maintain the consistency of the counting criteria, the author was the only one responsible for the counting works and no one else participated. Nevertheless, since this research involved techniques of mechanically recording, triangulation, and trick description, to ensure the trustworthiness, the influence of the bias of single evaluator could be minimized. However, this will also be listed as one of the limitations of this research.

Because a significant P value can only reflect that there is an observed difference and cannot reveal the size of the effect, reporting the effect size is necessary (Sullivan & Feinn, 2012). As suggested by numerous previous research, such as Cohen (1992), Dybå, Kampenes, and Sjøberg, (2006), and Erdfelder, Faul, and Buchner (1998), the information of effect size, as well as statistical power, should be provided; otherwise, the significant and insignificant results might be meaningless. Therefore, in this study, in addition to P values, the effect size (i.e., Cohen's *d*) was calculated to identify whether the significant results were substantial in the population. Moreover, to understand whether the small sample size (*N* = 32) was powerful enough to detect the substantial results, statistical power was also calculated. In this study, G*Power was used to calculate the effect size and power because it is a widely-used and professional program for power analysis (Erdfelder et al., 1998).

3.5.2 Qualitative Analysis

Thematic analysis was used for the qualitative data collected by videotaped/radio-taped classroom, observation notes, and interviews. For data management, all radio tracks (i.e., radio-taped classroom and interviews) were transcribed literatim and input into the NVivo software. All videos were divided into clips (10 minutes for each). As can be seen from Appendix E, a coding scheme based on the research purposes and the conceptual framework is provided. This coding scheme was used for analysing all the qualitative data as a first-round coding. Supplementary themes that emerged from the first-round coding were also added to the second-round coding.

When exploring the third research question (i.e., what might be the relationship between three classroom interactions and students' learning outcome development?), which is the focus of this research, eight representative focal samples were critically selected. Their performances were particularly focused when the researcher analysing the videotaped data. Among eight focal samples, four were used for probing effective classroom interactions for developing students' PS competencies, and another four samples were used for discovering effective classroom interactions for developing students' CT skills. The criteria for selecting focal samples were: 1) their performances in PS competencies/CT skills were at a relatively similar level in the pre-test/the first six days; 2) half of them have made significant progress in PS competencies/CT skills, while the other half have made little or no progress; 3) they were all grade-2 student; 4) the number of girls and boys should be the same. As a result, eight focal samples were selected. By comparing classroom interactions between those who have made progress in PS competencies/CT skills and those who have not, this research found several effective classroom interactions. Which is important that, although the researcher particularly focused on eight focal samples when analysing the videotaped data, other students' behaviours were also carefully analysed.

3.5.3 Trustworthiness

It is crucial to ensure the trustworthiness of an evidence-based research. Due to the reason that the philosophical worldviews and methodological assumptions are different among various research methods, the evaluation standards for different methods used to ensure the rigor of the research are also different (Anney, 2014). For example, the reliability, objectivity, and validity are usually considered by a quantitative research, while the qualitative researcher concern more about dependability, credibility, transferability and confirmability (Anney, 2014). Nevertheless, some scholars argued that there are specific trustworthiness standards for the mixed-method research (Cohen et al., 2011; Onwuegbuzie & Johnson, 2006). This research agreed with this argument, because the philosophical worldviews and methodological assumptions for mixed-method research are quite different from the pure qualitative or quantitative research. Next, based on Zohrabi's (2013) suggestions, how this mixed-method research addressed the trustworthiness will be explicit described.

Validity. "In the main, validity is concerned with whether our research is believable and true and whether it is evaluating what it is supposed or purports to evaluate." (Zohrabi, 2013, p. 258). This research validated the instrument and data in terms of content validity, internal validity, and external validity.

Firstly, for ensuring the content validity of each measurement, some strategies were adopted by this research. For example: A panel of experts in education and research was organized for evaluating the content validity of CSPSI (see details in the section 3.6.2); the scoring criteria of CTR as well as the interview protocol for students were carefully reviewed by the researcher and the teacher; the interview protocol for the teacher was piloted in the first-time interview and his comments on the interview questions were collected for improving the content validity of the protocol; the observation form was designed based on the conceptual framework which was reviewed by the principal supervisor of the researcher.

Secondly, as suggested by Merriam (1998), triangulation, member checks, and minimizing researcher's bias were used to address the internal validity which "is concerned with the congruence of the research findings with the reality" (Zohrabi, 2013, p. 258). For triangulation, this research adopted various data collection methods, including pre-/post-tests, rubric scoring, semi-structured interviews, and classroom observation. To answer each research question, at least two methods were used (see details in the section 3.2.3). Also, given that all student informants were at young age (i.e., from 7 to 9), information collected from them could be anamorphic due to their limited ability to express, so informants triangulation was adopted for minimizing such kind of biases. Moreover, scripts of the interview with the teacher were sent back to and reviewed by the teacher participant "to confirm the content of what they have stated during the interview encounter" (Zohrabi, 2013, p. 258). Furthermore, during the data collection and analysis process, the researcher tried to be critical and faithful, to remain as non-judgmental, and to insist on accurately perform the evaluation and honestly report the findings (Zohrabi, 2013). Although other methods could also be used for better ensuring the internal validity, such as long-term observation, peer examination, participatory/collaborative modes of research, due to the limitation of resources and some considerations about not letting students' study be negatively affected, this research was unable to adopt all suggested strategies.

Thirdly, "external validity is concerned with the applicability of the findings in other settings or with other subjects." (Zohrabi, 2013, p. 259). The findings of current study could be transferred to other settings, such as school settings, because the environment, the learning tool, the programming interface, the in-class curriculum, and other elements of the chosen summer camp, as well as the subjects, were not irreplaceable.

Reliability. Examining the reliability of quantitative data is relatively easy, such as conducting the internal consistency reliability analysis (calculating Cronbach's alpha coefficient) and the inter-rater reliability analysis (calculating Kappa coefficient), as this research has done. However, for qualitative data which in narrative form and subjective, "it is better to think about the dependability and consistency of the data" (Zohrabi, 2013, p. 259). In the current study, some strategies were adopted for addressing the dependability and consistency (i.e., internal and external reliability) of this study.

To increase the dependability of this study, three techniques suggested by Lincoln and Guba (1985) and Merriam (1998), were used. The first is thick description. This thesis explicitly explains the rationales of research design and methods used and clarifies as many aspects of this study as it can. Secondly, as mentioned before, various types of methods were adopted for triangulation. Thirdly, for the audit trail, all the data collection and analysis procedure are explicitly demonstrated in this thesis, which might be helpful for replicating the research.

To ensure the internal reliability of this study, all qualitative data from interviews and classroom observations, were mechanically recorded, which could be easily reanalysed or replicated (Zohrabi, 2013). As regarding of the external reliability, five important aspects of the inquiry, including "the status of the researcher, the choice of informants, the social situations and conditions, the analytic constructs and premises and the methods of data collection and analysis.” (Zohrabi, 2013, p. 260), were carefully considered and described in the corresponding parts of this chapter.

3.6 The Pilot Study: Validating CSPSI

The purpose of this pilot study was to examine the validity and reliability of SPSI when it was translated into the Chinese version and adapted for Chinese primary-level students. This section will first introduce the translation process and then discuss how the content validity of CSPSI was ensured. Samples and the data collection procedure will be presented, and the results of construction validity and internal consistency reliability will be offered.

3.6.1 Translation

Tian, Heppner, and Hou (2014) have translated PSI into Chinese Mandarin and validated it with Chinese people, though their study provided only 13 items and only some of the items overlapped with SPSI. Therefore, this study used those items having a Chinese translation and translated other items by using two professional and experienced translators with knowledge of problem-solving in education. After the initial translation, the two translators discussed and carefully determined whether the translated items were close to the English version and suitable for the children participants. Consequently, some wordings were fine-tuned. After that, the translated items were translated back into English by a professional translator. An English native speaker was invited to compare the translated-back version to the original version to determine whether the translated items were proper. Through all the steps above, the Chinese version of SPSI (CSPSI) was finalized and prepared for the pilot study (see Appendix C).

3.6.2 Content Validity

Because the original items of SPSI were for adults, the issue of whether the content of items was suitable for young children should be further checked. Firstly, a panel consisting of three experienced primary school teachers from the mainland of China was organized. They read all the items carefully and provided comments based on their experiences with children. They agreed that all the items were suitable for primary school students, but they suggested that the instruction of the test should be clearer. Therefore, the instruction part was refined. After that, 10 primary school students from mainland China were invited to take the trial test with the CSPSI. Students' perceptions of each item were obtained and recorded, then discussed by the panel. The panel members and the researcher then refined the wordings of some items and repeated the translate-back processes until the scale reached a good level of content validity.

3.6.3 Administration of Items to a Pilot Sample

The modified CSPSI of 16 items was tested by 118 primary students from three primary schools on the mainland of China by the snowball sampling method until a desirable number for covering different stratified groups was reached. The participants ranged in age from seven to nine years. Of these participants, 70 were girls and 48 were boys.

3.6.4 Data Collection

The CSPSI and consent letters were delivered to each student participant by the head teacher. Firstly, the teacher read aloud the instructions and each of the 16 items, one at a time. Participants rated the extent to which they agreed with the statements. A 6-point Likert scale (From 1- *Strongly disagree* to 6- *Strongly agree*) was used for rating.

3.6.5 Factor Structure

To identify the latent factor structure of the scale, an Exploratory Factor Analysis (EFA) was conducted using Principal Components Factor (PCF) analysis to investigate the dimensional structure of CSPSI. The Kaiser-Meyer Olkin (KMO) measure of sampling adequacy was .811 and Bartlett’s test of sphericity was statistically significant, $\chi^2 = 2469.12$, $df = 91$, $p < .001$, which supported the appropriateness for conducting the EFA (Hair, 2014). As suggested by Hair (2014) and Netemeyer and colleages (2003), the varimax orthogonal rotation was used for EFA; items with a pattern coefficient greater than .50 were retained.

Based on the principle of “eigenvalue-over-one,” a two-factor solution with 14 items was suggested by EFA. Two items (i.e., Q8 and Q11) were excluded because their pattern coefficient was under .50. The 14 retained items accounted for 80.54% of the total variance (a shown in Table 5. The factor structure could be summarized as follows:

Factor 1: Q1, Q2, Q3, Q4, Q5, Q6, Q7

Factor 2: Q9, Q10, Q12, Q13, Q14, Q15, Q16

3.6.6 Confirmatory Factor Analysis (CFA)

To identify the model fitness of the two-factor model suggested by EFA, a CFA was conducted by using Structural Equation Modelling (SEM) with RStudio. The results showed that model Chi-square test $\chi^2(28, N = 118) = 32.69, p = .247$, suggesting that the model fit perfectly. Meanwhile, other goodness-of-fit indices such as *CFI* = .998, *TLI* = .996, *RMSEA* = .033, and *SRMR* = .047 suggest the initial two-factor model has a good fit to the data. Therefore, the two-factor structure of the scale was identified and accepted.

Table 5

Results of Exploratory Factor Analysis

KMO and Bartlett's Test		
Kaiser-Meyer-Olkin Measure of Sampling Adequacy	.811	
Bartlett's Test of Sphericity		
Approx. Chi-Square	2469.116	
df	91	
Sig.	.000	
Extraction Sums of Squared Loadings	Component 1	Component 2
Total	5.754	5.522
% of Variance	41.099	39.445
Cumulative %	41.099	80.544
	Rotated Factor Pattern Coefficients	
Items	Factor 1	Factor 2
Factor 1		

1. When my first efforts to solve a problem fail, I become uneasy about my ability to handle the situation.	.838	
2. I have the ability to solve most problems even though initially no solution is immediately apparent.	.940	
3. Many of the problems I face are too complex for me to solve.	.797	
4. When I make plans to solve a problem, I am almost certain that I can make them work.	.939	
5. Given enough time and effort, I believe I can solve most problems that confront me.	.911	
6. When faced with a novel situation, I have confidence that I can handle problems that may arise.	.912	
7. I trust my ability to solve new and difficult problems.	.931	
Factor 2		
9. After following a course of action to solve a problem, I compare the actual outcome with the one I had anticipated.		.858
10. When I have a problem, I think of as many possible ways to handle it as I can until I can't come up with any more ideas.		.932
12. When confronted with a problem, I stop and think about it before deciding on a next step.		.882
13. When making a decision, I compare alternatives and weigh the consequences of one against the other.		.928
14. I try to predict the result of a particular course of action.		.891
15. When thinking of ways to handle a problem, I seldom combine ideas from various alternatives to arrive at a workable solution.		.860
16. When confronted with a problem, I usually first survey the situation to determine the relevant information.		.903

3.6.7 Naming Factors

Because the results of both EFA and CFA suggested a two-factor structure of CSPSI, which is consistent with the results of Maydeu-Olivares and Zurilla (1997), the two factors were named as the same as in the previous study. Factor 1 containing Q1, Q2, Q3, Q4, Q5, Q6, and Q7, concerning self-efficacy when solving problems, was named **Problem-Solving Self-Efficacy** (PSSE). Factor 2 comprised seven items (Q9, Q10, Q12, Q13, Q14, Q15, and

Q16) regarding the problem-solving skills, and was named **Problem-Solving Skills** (PSS). These two concepts were already introduced in the conceptual framework.

3.6.8 Internal Consistency Reliability

The results indicated satisfactory internal consistency reliability of the CSPSI. The Cronbach's alpha coefficient was .88 for the total score of CSPSI. The Cronbach's alpha coefficient of the two subscales of the CSPSI was .95 (Factor 1) and .96 (Factor 2). The range of corrected item-total correlations for the 14 items was from .42 to .58. The inter-item correlations ranged from -.16 to .98 with a mean of .35. All of these correlation coefficients were within the moderate range. According to Clark and Watson (1995), the results suggested that the CSPSI possessed high internal consistency. With a sample size of 118, the Pearson correlations of the total score of CSPSI and its two subscale scores were: Factor 1: $r = .75$, $p < .001$; and Factor 2: $r = .64$, $p < .001$.

3.7 Ethical Issues

This research has gained the approval for conducting research and collecting data from the Human Research Ethics Committee (HREC) of EdUHK. All data collection procedures were claimed in the ethical review report. Because the student participants were all younger than 18, according to the requirements of HREC, the consent letters containing statements about the research purposes, data collection procedure, potential risks, confidentiality of data, and other research information were delivered to students, their parents, and the organization (i.e., The Children's Palace) when participants were recruited. The signed consent letters were collected. During the research, participants' rights were guaranteed. All data collected were strictly confidential, so that only the researcher could access the database.

3.8 Limitations

Although the methodology of this research had been justified in this chapter, some limitations of the methodology should be acknowledged and discussed.

First of all, the sample size (*N* = 32) was small. The small sample size was due to several reasons. First, because the classroom had only 10 sets of robot kits and 10 laptops, the size of each class had to be small to ensure that each student has their own kit and laptop. Second, during the process of recruiting the students, many of their parents were concerned about the class size and said that they preferred the class size to be less than 10, as they thought that a larger class might decrease the quality of their children's learning. Third, the teacher preferred a smaller class size because robotics learning involves lots of classroom interactions; thus, one teacher could not pay attention to many students. Fourth, from the perspective of the researcher, the teacher, as a crucial variable directly influencing classroom interactions and students' learning outcomes, should be controlled; this meant that the same teacher would provide instruction for every class. Also, to ensure that students had one lesson every two days, only four classes could be arranged (i.e., morning and afternoon sessions every day); therefore, only four classes with 40 students at most could be involved in this research. Fifth, as stated previously, there were some precedents for using a small sample size in RE research. For example, in Xia and Zhong's (2018) reviewed study, 47.62% of the literature they reviewed had a sample size smaller than 40. In the literature reviewed by the author, some quantitative research and mixed-method research was conducted with samples less than 40 (see examples for quantitative research: Kucuk & Sisman, 2017; Lee et al., 2013; Liu et al., 2013; for mixed-method research: Barak & Assal, 2018; Zhao, 2018). As a result, this research acknowledged that the small sample size was a limitation and weakness but, due to considerations above, it was decided to accept the small sample size and to use various techniques (such as conducting a post hoc power test, engaging in investigator triangulation, and focusing more attention on qualitative data) to address this problem.

Secondly, only one teacher was involved, because teacher's performances and behaviours might directly influence the classroom interactions and students' learning outcomes. In order to control the variables influencing the research findings, this research only involved one teacher. Apart from that, the purpose of this research was not to compare different teachers. Because of this, all the students were taught by the same teacher. Nonetheless, there might be a problem that the information collected from 32 students and one teacher were unbalanced, such as the data in the interviews. For addressing this problem, this research had conducted interviews with the teacher for two times, with the length of each time being more than one hour. In addition, the data collected from the teacher will be presented adequately in the finding chapter of this thesis.

Lastly, the duration of this research was relatively short. As reviewed before, many of the previous RE researches were conducted in the informal learning environment, such as the summer camp adopted by this research, because of its short-term duration as well as the minimum curriculum design and teacher training (Mubin et al., 2013). In mainland China, the short-term summer camps are an important form of carrying out RE. The author believes that, the potential results of this short-term study are still significant to the field of RE. Despite the short duration of this study, students' learning was intensive (i.e. three lessons per week). Thus, there was 18-hour study time for the researcher to observe the changes of students. Nevertheless, the future research could focus on students' performance in the long-term robotics learning.

Chapter 4: Quantitative Results

4.1 Problem-Solving Competencies

Students' problem-solving competencies were measured by CSPSI. Both students and the teacher were required to complete the pre- and post- tests for problem-solving competencies. The following sections will present the results of students' perceptions of their own problem-solving competencies and the teacher's perception of students' problem-solving competencies. In addition, these two sets of data (i.e., data from students and the teacher) will be merged (by counting the means) and the results of the merged data will be presented.

4.1.1 Students' Perceptions of Their Own Problem-Solving Competencies

To compare students' problem-solving competencies before and after the summer camp, Wilcoxon tests were conducted with the pre-/post-test scores of students' perceptions. As shown in Table 6, the results of Wilcoxon tests indicated that students' problem-solving competencies (i.e., the sum score of PSSE and PSS) were significantly increased from the pre-test (M_{pre} = 9.55, Mdn_{pre} = 9.64) to the post-test (M_{post} = 10.47, Mdn_{post} = 10.50), with z = -3.21, T = 435.5, and p = .001. Also, with a medium-level effect size (d = .69) and high power (= .98), the difference between the sum scores of the pre- and post-tests was substantial in the population and the sample size (N = 32) was powerful enough to detect the difference. Specifically, for the PSSE subscale, the increase was nonsignificant (M_{pre} = 4.81,

M_{post} = 5.20, Mdn_{pre} = 4.86, Mdn_{post} = 5.29, z = -1.93, T = 326, p = .054 > .05), while for the PSS subscale, the increase was significant (M_{pre} = 4.73, M_{post} = 5.27, Mdn_{pre} = 5, Mdn_{post} = 5.29, z = -2, T = 309, p = .046 < .05). In terms of the effect size and power of the above results (i.e.: for PSSE and PSS subscales), for the result of PSSE, the small effect size (d = .35) indicated that the nonsignificant difference might not be due to the small sample size, while the low level of power (= .66) suggested that it was hard to detect the difference between the pre- and post-tests of PSSE. Nevertheless, for the result of the PSS subscale, the medium-level effect size (d = .50) and high power (= .92) indicated that the significant difference was substantial in the population and that the sample size (N = 32) was powerful enough to detect the difference.

Table 6

Results of Wilcoxon Test for Student's Perception of Their Own Problem-solving Competencies (N = 32)

	Pre-test			Post-test			*z*	*T*	*P*	*d*	Power
	M	*SD*	*Mdn*	*M*	*SD*	*Mdn*					
PSSE	4.813	.934	4.857	5.196	.611	5.286	-1.925	326	.054	.349	.660
PSS	4.732	.985	5.000	5.272	.311	5.286	-1.995	309	.046	.501	.920
Total	9.545	1.199	9.643	10.469	.682	10.500	-3.210	435.5	.001	.692	.997

Overall, based on students' perceptions of their own performances, there was a positive shift in problem-solving competencies during the robotics summer camp. Additionally, their problem-solving skill was mainly enhanced, while problem-solving self-efficacy nonsignificantly increased.

4.1.2 The Teacher's Perceptions of Students' Problem-Solving Competencies

Similar to reporting students' perception of their problem-solving competencies in the last section, the teacher's perceptions of students' problem-solving competencies before and after the summer camp were compared by using Wilcoxon tests. As shown in Table 7 the results of the Wilcoxon tests indicated that students' problem-solving competencies (i.e., the sum score of PSSE and PSS) significantly increased from the pre-test (M_{pre} = 8.60, Mdn_{pre} = 8.71) to the post-test (M_{post} = 10.02, Mdn_{post} = 10.36), with z = -3.21, T = 526, and p < .001. Specifically, for the PSSE subscale, the increase was significant (M_{pre} = 4.31, M_{post} = 4.84, Mdn_{pre} = 4.36, Mdn_{post} = 4.93, z = -3.67, T = 435, p < .001), and for the PSS subscale, the increase was also significant (M_{pre} = 4.29, M_{post} = 5.18, Mdn_{pre} = 4.64, Mdn_{post} = 5.64, z = -5.09, T =528, p < .001). In terms of the effect size and power of the above results, for all results, the effect sizes were at moderate to high levels and the powers were high, which indicated that the significant results were substantial in the population and that the sample size (N = 32) was powerful enough to detect the significant results.

Table 7

Results of Wilcoxon Test for the Teacher's Perceptions of Students' Problem-Solving Competencies (N = 32)

	Pre-test			Post-test			*z*	*T*	*p*	*d*	Power
	M	*SD*	*Mdn*	*M*	*SD*	*Mdn*					
PSSE	4.313	0.655	4.357	4.839	0.334	4.929	-3.672	435	.000	.711	.998
PSS	4.286	0.853	4.643	5.179	0.898	5.643	-5.089	528	.000	4.934	1
Total	8.598	0.851	8.714	10.018	0.958	10.357	-4.905	526	.000	2.015	1

In summary, based on the teacher's perceptions of students' performance, there was a positive shift of problem-solving competencies during the robotics summer camp. Additionally, both problem-solving self-efficacy and skill were better after the robotics summer camp than before it.

4.1.3 Correlation between Students' and the Teacher's Perceptions

Because this study measured not only students' perception of their own problem-solving competencies but also the teacher's perception of every student's problem-solving competencies, it is meaningful for us to know whether these two aspects of data were correlated. Therefore, Spearman's correlation analysis was conducted for these two sets of data. As shown in Table 8, in the pre-test, the correlations between the students' and teacher's scores for two subscales and for sum scores were significantly positive, with *rs*(30) = .849, .487, and .691, $p < .001$. For the post-test data, the correlation for the PSS subscale was significantly positive (*rs*(30) = .90, *p* < .001), while for the PSS subscale, with *rs*(30) = -.37 (*p* = .037), a low-level negative correlation was determined. For the sum scores of the post-test, the correlation was rare (*rs*(30) = .07, *p* = .722). Additionally, comparing the means of the two perceptions, for all cases, the results indicated that the students' subjective ratings were higher than the teacher's external ratings.

Table 8

Results of Spearman's Correlation: Students' Perception and the Teacher's Perception (N = 32)

	Pre-Test			Post-Test		
	Student (*M*)	Teacher (*M*)	*rs*	Student (*M*)	Teacher (*M*)	*rs*
PSSE	4.813	4.313	.849**	5.196	4.839	.899**
PSS	4.732	4.286	.487**	5.272	5.179	-.370*
Total	9.545	8.598	.691**	10.469	10.018	.066

Note.
** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

Overall, the results above suggested that, to some extent, there might be inconsistencies between the students' and the teacher's ratings for problem-solving competencies; however, it is hard for us to know which side is the "fact" only by their scores, as either subjective or objective/external measures could contain bias (Schneider et al., 2019). Therefore, it is reasonable to report and analyse the merged scores of both sides. Also, in addition to quantitative data, qualitative information would be crucial to drawing a whole picture of students' problem-solving competencies.

4.1.4 Merged Data of Problem-Solving Competencies

Two sets of data were merged by simply calculating their means; the merged data were then tested by the Wilcoxon test again. As shown in Table 9, the results of the Wilcoxon tests indicated that students' problem-solving competencies (i.e., the sum score of PSSE and PSS) significantly increased from the pre-test (M_{pre} = 9.07, Mdn_{pre} = 9.21) to the post-test (M_{post} = 10.24, Mdn_{post} = 10.36), with z = -4.63, T = 484, and p < .001. Specifically, for the PSSE subscale, the increase was significant (M_{pre} = 4.56, M_{post} = 5.02, Mdn_{pre} = 4.57, Mdn_{post} = 5.07, z = -2.69, T = 408, p = .007), while for the PSS subscale, the increase was also significant (M_{pre} = 4.51, M_{post} = 5.23, Mdn_{pre} = 4.75, Mdn_{post} = 5.36, z = -4.94, T =528, p < .001). In terms of effect size and the power of the above results, for all results, the effect sizes were at moderate to high levels and the powers were high, which indicated that the significant results were substantial in the population and the sample size (N = 32) was powerful enough to detect the significant results.

Table 9

Results of Wilcoxon Test for Merged Data of Problem-Solving Competencies (N = 32)

	Pre-test			Post-test			*z*	*T*	*p*	*d*	Power
	M	*SD*	*Mdn*	*M*	*SD*	*Mdn*					
PSSE	4.563	0.766	4.571	5.018	0.454	5.071	-2.694	408	.007	.511	.934
PSS	4.509	0.847	4.750	5.225	0.431	5.357	-4.944	528	.000	1.406	1
Total	9.071	0.947	9.214	10.243	0.615	10.357	-4.625	484	.000	1.362	1

In summary, based on the merged data, there was a positive shift in students' problem-solving competencies during the robotics summer camp. Additionally, both problem-solving self-efficacy and skill were better after the robotics summer camp than before it.

4.1.5 Summary

Synthesizing all the results above, three important results about students' problem-solving competencies were found. Firstly, students' perceptions and the teacher's perceptions might be inconsistent; students tended to score themselves higher than the teacher did. Secondly, based on different perceptions and emerged data, the results of comparing the pre- and post-tests were generally consistent, which indicated a significant positive shift of students' problem-solving competencies during the robotics summer camp. Thirdly, most of the results above were substantial in the population, and the sample size (N = 32) was powerful enough to detect significant results.

4.2 Computational Thinking Skills

Students' computational thinking skills were measured by using the CTR (Computational Thinking Rubric). Both the teacher and the researcher scored students' performances in every class. The following sections will present the correlation of the teacher's and researcher's scores and the comparison of the scores of the first six days and the last six days.

4.2.1 Measure of Inter-Rater Agreement between the Teacher's and Researcher's Scores

Because two raters (i.e., the teacher and researcher) were involved in measuring students' CT performances by using CTR, we must know the degree of agreement among raters (i.e., inter-rater reliability). As suggested by Cohen (1960), inter-rater agreement should be measured by Cohen's Kappa test. Each domain of the rubric was tested, respectively. As shown in Table 10, the Kappa coefficients of six CT domains suggested that the levels of agreement between the two raters were from moderate to strong ($.54 < \kappa < .92$, $p < .001$). Namely, when scoring students' computational thinking level, the teacher and the researcher showed consistent opinions. It is reasonable to merge two raters' scores. Because the two raters were objective (it herein means "external"), this study did not separately analyse each rater's scores. The emerged data counting by mean of two raters' scores was used for further analysis.

Table 10

Results of Cohen's Kappa Test: Two Raters' Agreement Measure

	C1 [a]	C2 [a]	C3 [a]	C4 [a]	C5 [a]	C6 [a]
Kappa coefficient	.669	.923	.857	.787	.544	.797
P value	.000	.000	.000	.000	.000	.000

Note.

[a] C1 to C6 refer to six domains of CTR: C1- Formulating problems, C2- Abstraction, C3- Logical thinking, C4-Using algorithms, C5- Analysing and implementing solutions, C6- Generalizing and problem transfer.

4.2.2 Mean Difference between CT Scores of the First Six Days and the Last Six Days

Based on the merged data of the CT scores, students' CT skill during the first six days was compared to that of the last six days by using Wilcoxon tests. As shown in Table 11, the results of the Wilcoxon tests indicated that students' CT skill (i.e., sum score of six domains) significantly increased from the first six days (M_{first} = 7.33, Mdn_{first} = 7) to the last six days (M_{last} = 9.39, Mdn_{last} = 8.96), with z = -4.94, T = 528, and p < .001. Specifically, for all six domains of CTR, the increases were significant. In terms of the effect size and power of the above results, for all the results, the effect sizes were at moderate to high levels and the powers were high, which indicated that the significant results were substantial in the population and that the sample size (N = 32) was powerful enough to detect the significant results.

Table 11

Results of Wilcoxon Test for Computational Thinking Skills (N = 32)

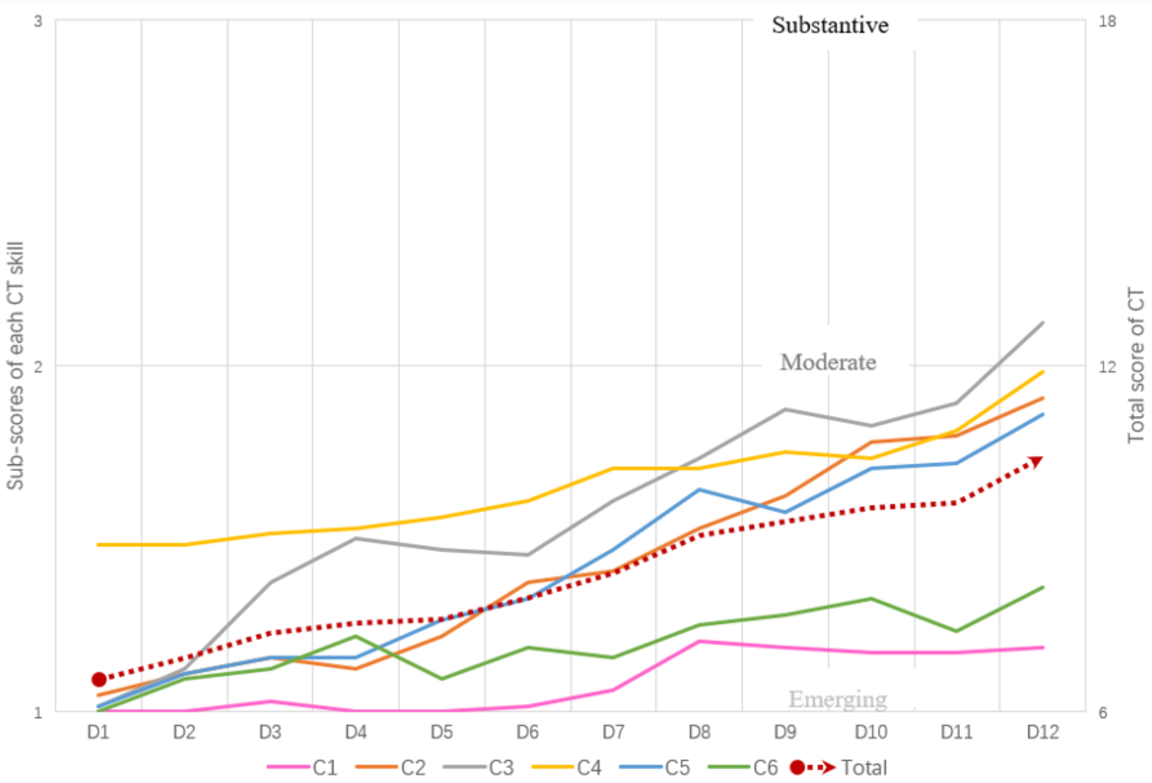
	First-six-day			Last-six-day			*z*	*T*	*p*	*d*	Power
	M	*SD*	*Mdn*	*M*	*SD*	*Mdn*					
C1 [a]	1.008	.033	1.000	1.164	.233	1.083	-3.638	153	.000	.692	.997
C2 [a]	1.172	.257	1.167	1.674	.367	1.667	-4.846	495	.000	1.980	1
C3 [a]	1.323	.442	1.167	1.844	.539	1.792	-4.756	464	.000	1.364	1
C4 [a]	1.531	.428	1.458	1.781	.452	1.750	-3.298	348	.001	.679	.996
C5 [a]	1.172	.230	1.083	1.661	.359	1.667	-4.669	459	.000	1.392	1
C6 [a]	1.120	.167	1.000	1.268	.350	1.167	-2.794	247	.005	.552	.963
Total	7.326	1.229	7.000	9.393	1.716	8.958	-4.938	528	.000	2.507	1

Note.
[a] C1 to C6 refer to six domains of CTR: C1- Formulating problems, C2- Abstraction, C3- Logical thinking, C4-Using algorithms, C5- Analysing and implementing solutions, C6- Generalizing and problem transfer.

Overall, there was a positive shift in computational thinking skill during the robotics summer camp. Additionally, all six CT skills (i.e., formulating problems, abstraction, logical thinking, using algorithms, analysing and implementing solutions, and generalizing and problem transfer) were better on the last six days than on the first six days.

4.2.3 Trending of Students' Computational Thinking Development

As mentioned previously, students' computational thinking skill improved during the robotics summer camp. Because students' performances were scored day-by-day, we could further explore the trending of students' computational thinking development. As shown in Figure 4, students' computational thinking (i.e., total score, the red dashed line) gradually increased from the emerging level to the moderate level; however, the score did not reach the moderate level until the end of the summer camp. Students' logical thinking (i.e., C3, the grey line) was enhanced most significantly and surpassed the moderate level on the last day. Students' progress in terms of their abilities for abstraction (i.e., C2, the orange line) and analysing and implementing solutions (i.e., C5, the blue line) were also remarkable. However, students' abilities to formulate problems (i.e., C1, the pink line) and generalize and problem transfer (i.e., C6, the green line) were limitedly enhanced. As for students' algorithm use (i.e., C4, the yellow line), the starting level was higher than that of other CT components, while the trending increased slightly.



Notes.

a. D1 to D12 refer to Day 1 to Day 12.

b. C1 to C6 refer to six CT skills.

c. "Total score" means the sum score of six components.

d. In CTR, 1 means *Emerging* level, 2 means *Moderate* level, and 3 means *Substantive* level. Therefore, the range of each sub-score is from 1 to 3; and, for the total score, the range is from 6 to 18.

Figure 4. The Trending of Students' Computational Thinking Skills (Day by Day).

4.2.4 Summary

A synthesis of all the results above found four important results regarding students' computational thinking skills. Firstly, the results scored by the teacher were consistent with the scores from the researcher. Secondly, the results stemming from the merging of the teacher's and the researcher's scores indicated that there were significant differences between students' first-six-days performances and their last-six-days performances. Namely, students' computational thinking skill was significantly enhanced during the summer camp. Thirdly, all the results above were substantial in the population, and the sample size (*N* = 32) was powerful enough to detect significant results. Fourth, according to the trending figure based on students' day-by-day performance, students' computational thinking skills increased gradually but did not reach the moderate level until the end of the summer camp.

4.3 Correlation between PS competencies and CT skills

A Spearman correlation analysis was conducted to identify the correlation between students' post-test scores for problem-solving competencies and last-six-days scores for computational thinking skills. The result indicated a significant positive correlation, with *rs*(30) = .64 and *p* < 0.001.

4.4 Time Spent on Students' Interactions

4.4.1 Time Spent on Students' Interactions with the Teacher, Peers, and The Robot

Students' time spent on interactions with the teacher, peers, and the robot was counted, respectively (see Figure 5). The mean time spent on Student-Teacher interactions was about 289 minutes (25%), which was the least part of the students' interactions. The most time spent was on Student-Student interactions, which was about 484 minutes (41%). Student-Robot interactions accounted for about 397 minutes (34%). Which should be noted is that, three types of interactions could possibly be overlapped; for example, students interacted with the robot while interacting with their peers.

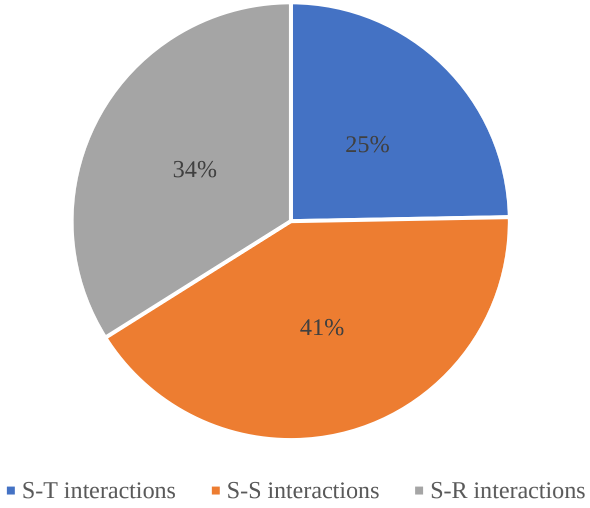


Figure 5. Percentage of Time Spent on Three Classroom Interactions.

4.4.2 Relationships between Classroom Interactions and Students' Problem-Solving Competencies

The relationships between the increase in students' problem-solving competencies (the mean difference between the pre- and post-test) and time spent on three classroom interactions were examined by Spearman's correlation analysis. As shown in Table 12, the increase in students' PSSE was positively correlated with time spent on S-S interaction ($rs(30) = .55$, $p = .001$). Meanwhile, the increase in students' PSS was positively correlated with time spent on S-T ($rs(30) = .62$, $p < .001$) and S-R interactions ($rs(30) = .40$, $p = .024$) at a significant level ($p < .05$). Lastly, the increase in students' problem-solving competencies (i.e., PSSE and PSS together) was positively correlated to time spent on S-T interactions (*rs*(30) = .42, *p* = .015), S-S interactions (*rs*(30) = .63, *p* < .001), and S-R interactions (*rs*(30) = .46, *p* = .008).

Table 22

Results of Spearman's Correlation Analysis: Time Spent on Classroom Interactions and Mean Difference of Students' Problem-solving *Competencies between Pre- and Post- Tests (N = 32)*

	S-T interactions	S-S interactions	S-R interaction
PSSE	.155	.554**	.336
PSS	.619**	.222	.399*
Total	.424*	.628**	.462**

Note.
** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

4.4.3 Relationships between Classroom Interactions and Students' Computational Thinking Skills

The relationships between the increase in students' computational thinking skill (the mean difference between the first six days and the last six days) and time spent on interactions were examined by Spearman's correlation analysis. As shown in Table 13, the enhancement of the skill of abstract was positively correlated with time spent on S-T interaction (*rs*(30) = .38, *p* = .033) and S-S interaction (*rs*(30) = .35, *p* = .048). For logical thinking skills, time spent on S-S interaction was positively correlated to its increase (*rs*(30) = .46, *p* = .008). For the skill of analysing and implementing solutions, its increase was positively correlated to time spent on S-T interaction (*rs*(30) = .43, *p* = .014) and on S-S interaction (*rs*(30) = .36, *p* = .046). For the total score of computational thinking, the enhancement was positively correlated to time spent on S-T interactions (*rs*(30) = .60, *p* < .001), S-S interactions (*rs*(30) = .64, *p* < .001), and S-R interactions (*rs*(30) = .51, *p* = .003).

Table 33

Results of Spearman's Correlation Analysis: Time Spent on Classroom Interactions and Mean Difference between Students' CT Skill between the first six days and the last six days (N = 32)

Domains of CTR	S-T interactions	S-S interactions	S-R interaction
Formulating problems	.158	.093	.195
Abstraction	.378*	.352*	.215
Logical thinking	.150	.461**	.267
Using algorithms	.214	.174	.286
Analysing and implementing solutions	.428*	.355*	.240
Generalizing and problem transfer	.124	.001	.015
Total	.595**	.640**	.513**

Note.
** Correlation is significant at the 0.01 level (2-tailed).
* Correlation is significant at the 0.05 level (2-tailed).

Chapter 5: Qualitative Findings

Every class of the summer camp started with the teacher’s introduction to some basic concepts of building and programming a robot corresponding to the teaching purposes and lesson tasks (Stage 1 Introduction). Then, students started to build a robot with their own robot kit by following the teacher’s demonstration (Stage 2 Building robot). Programming and debugging for the robot were carried out by the students themselves (Stage 3 Programming and debugging). If there were any problems, students could ask their peers or the teacher for help or they could handle the situation by themselves. If anyone completed the task (i.e., making a vehicle), the teacher commented on the handiwork by observing the running of the robot and asking students about the logic behind making such a robot (Stage 4 Display and interpret).

According to the classroom observation, students’ PS competencies were involved mainly in Stages 2 and 3. There were two types of problems that students might encounter: problems with the building and problems with the programming. Students’ computational thinking was involved mainly in Stages 3, especially when they were debugging.

In this chapter, by reporting the qualitative data from the classroom observation and semi-structured group interviews, the following issues will be narrated thematically:

a) the features of students’ problem-solving in robotics learning, the evidence of students’ problem-solving competencies, the effective classroom interactions in developing problem-solving competencies, and

b) the features of students’ computational thinking in robotics learning, the evidence of students’ computational thinking skills, and the effective classroom interactions in developing computational thinking skills.

5.1 Students’ Learning Outcome: Problem-Solving Competencies

In terms of students’ PS competencies in the robotics learning environment, three main features were found in this research: Two kinds of problems (i.e., building problems and programming problems), a five-stage circle of PS process, and intuitively coming up with solutions. Also, some evidence of students’ PS competencies was found. Moreover, effective classroom interactions (i.e., S-S, S-T, and S-R) in students’ CT development will be presented.

5.1.1 Features of Students’ Problem-Solving in Robotics Learning

Two kinds of problems. Unlike the solving of daily personal problems or academic problems (e.g., mathematics or calculating problems), there were two typical types of problems in robotics learning: problems with *building* a robot (building problems) and problems with *programming* a robot (programming problems). Building problems are tangible and touchable, and students can tackle them visually; meanwhile, programming problems, to some degree, are nonobjective and students usually solved them through logical thinking or by transforming them into tangible forms.

Building problems were about how to select appropriate elements and assemble them to build a robot that could complete the lesson task. For example, in the Lesson 9, the lesson task was making a “walking” robot. Since the walking robot should “walk” like a human, there were several building problems that students had to deal with; for example, how to keep the balance when the robot is walking? Where to place the Intelligent Brick? How to move the robot’s “legs” by using different kinds of gears? When considering how to solve these building problems, students had tangible objectives, such as the robotics elements, to think with.

Programming problems were about how to input a series of commands to the robot and make the robot moves as expected. For example, on the third day of class (Lesson 3: Make your vehicle 2 (Go and return)), students were required to create a vehicle that not only could move forward but also return to the starting point. There was a typical problem in designing the program: how to make the vehicle return (turn 180°). Through guidance from the teacher, students already knew how to make the vehicle turn (e.g., set a forward function for one motor and set a backward function for another motor). However, they had to find a solution for turning 180° in programming. How would they solve this problem? They usually transformed this problem into a tangible form. For example, a student (D02) used the debugging function of the Intelligent Brick and tried using a different length of time for motor running. She found that, when the time was set at 10 seconds, the vehicle turned less than 180°, while when the time was set at 15 seconds, the vehicle turned over 180°. Then she tried 13 seconds and the vehicle turned exactly 180°. Consequently, the student put this data obtained from tangible tests into her programming and successfully completed the task. Another student (C02) adopted a different solution. Using a stopwatch, he measured the time consumed when a vehicle turned 360°. The time was 28 seconds, so he estimated that the vehicle needed 14 seconds to turn 180°. However, after putting “14 seconds” into his program, he noticed that the vehicle turned over 180°. “Why did the vehicle turn over 180°?” He was confused and ran the program on his vehicle again and again. Finally, through repeated observation, he found that “my car is running faster and more heavily than those of other classmates, so when it stops, it will turn continuously for a little while.” Although he had little knowledge about inertia force, he discovered this principle from observation. After tuning the parameter of his program, he successfully completed the task through the tangible trial-and-error process. As can be seen from examples above, robot programming is different from other types of programming, such as software programming, which is based mainly on algorithm design. It naturally needs to transform the programming or algorithm problem into a tangible form, especially for primary school children.

Problem-solving process: a five-stage cycle. Children's PS process generally followed a five-stage cycle (see Figure 6): identify the problem, come up with solution(s), make a decision, carry out the solution, and evaluate the solution. If the selected solution failed, the student went back to the first stage and completed the following stages until the problem was solved.

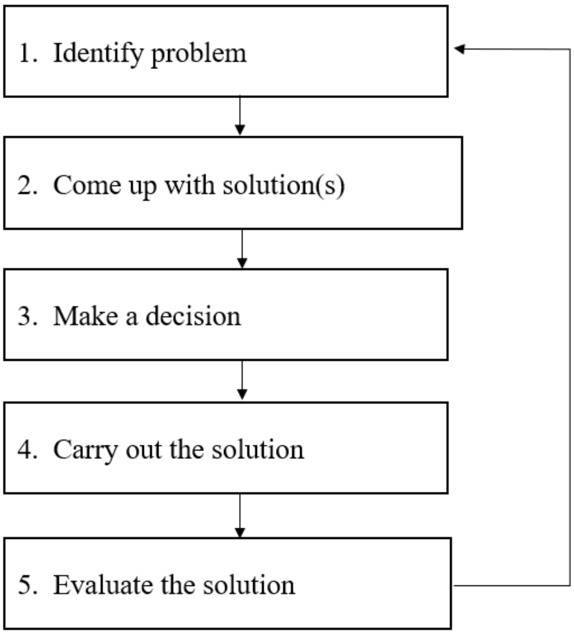


Figure 6. Students' Problem-Solving Process in Robotics Learning.

For example, in Lesson 6 (Make your vehicle 5), students were required to make an off-road vehicle with caterpillar bands. Because the number of caterpillar bands in the kit boxes was limited, the students had to resolve a building problem: how to use the limited caterpillar bands to build the off-road vehicle. A student (A05)'s PS process was as follows:

He firstly collected all the bands from his kit box, then estimated the length of the band he needed. Therefore, he identified that the lack of bands was a problem, as he did not have enough bands to make a caterpillar (Stage 1: identify the problem). The first solution he thought up was to borrow other students' bands (Stage 2: come up with a solution), so he decided to ask peers for more bands (Stage 3: make a decision). He asked around (Stage 4: carry out the solution) but found that this solution failed because there were no extra bands for him (Stage 5: evaluate the solution). Then, he backed up to reviewing the problem he had identified (i.e., the lack of bands) and revised his idea. He said, "I should think about how to use these [limited] bands to build my vehicle." (Stage 1). Therefore, he came up with a second solution: Make a smaller vehicle (Stage 2). However, he found that with the smaller size, there was not enough space to place the Intelligent Brick (Stages 3 to 5). Hence, he backed up again to review the problem he had identified (i.e., how to use limited bands to build the vehicle). This time, he did not modify the problem (Stage 1) but, rather, came up with a third solution: Make the Intelligent Brick "stand" on the vehicle so that the brick could also be settled on a smaller vehicle with limited bands (Stage 2). This time, the solution he came up with, decided on, and carried out proved to be successful (Stages 3 to 5).

The five-stage PS circle could be clearly observed in the example of student A05. This process starts at the stage of identifying the problem, where students define and formulate the problem they will solve. Based on the PS goal, students come up with one or more solution(s) and make a decision regarding which solution to carry out. After carefully carrying out the selected solution, students evaluate the outcome of the solution; if the solution fails, they go back to review and repeat the previous PS stages until the problem is successfully solved.

The *circled* process is the key feature of students' problem-solving because it represents a repeated process of trial-and-error. From the interview with the teacher, this feature can also be noticed:

"Among those who can successfully solve a problem, they usually tried to solve them through a trial-and-error process. They thought up some solutions and tried until they found the effective one."

Also, in terms of students' views of their own experiences, when the selected solution failed, they tried to find another way to solve the problem. For example, when asked, "What if you found that the solution did not work?," they responded:

Student B06: "I found another way to solve the problem."

Student A07: "Change to another way and try again."

Student D08: "Rethink my plan to see where the problem was."

Moreover, they did not think that they could always solve problems in a one-off manner:

Student B06: "Some simple problems I solved immediately, but for some difficult problems, I had to try two or more times."

Student B07: "Yes, when solving difficult problems, I had to try several times."

Student B08: "With luck, I can solve some problems immediately, but most of the time I cannot."

It should be noted that they did not always come up with a solution independently. Sometimes, they observed peers' solutions and were inspired by others' ideas, or directly sought help from peers and the teacher, particularly in the early classes. Nonetheless, in the later classes, their independence in generating solutions increased. Also, the interactions between students and their peers, the teacher, and the robot played significant roles in this enhancement, which will be interpreted in a later section.

Intuitively generate the solution(s). Generally, children created solutions in an intuitive manner. Namely, when they explained how a certain problem was solved, most of them could hardly elaborate on it.

For example, when students were asked "How did you come up with the solution(s) to problems?," their responses were obscure:

Student B01: "There was a solution in my mind, so I tried it and was successful."

Student B02: "I don't know how. I just solved it."

Student B03: "I thought it would work, and it did."

Because they were too young to express their ideas clearly and logically, the teacher's observation of this issue could provide some third-party information. For example, when he was asked, "Have you ever noticed any features of students' problem-solving?," he answered:

"…Maybe for them, at such a young age, it is hard to create a logical design or plan for solving the problem. According to my teaching experience, they thought up a solution usually by intuition, without a clear reason or rationale. Most of them had little knowledge about science, engineering, math, etc., but we found that they solved problems without using a specific concept or theory. For example, in the second lesson, they were required to create a vehicle that could move forward. I taught them only the basic structure of making a vehicle and told them to make one that could move as fast as possible. They came up with several solutions, such as using bigger wheels, removing the tire of the wheel [to reduce the force of friction], making a lighter car, or using a bigger gear to connect to a smaller gear. When I asked one of them why he removed the tire of the wheel, he told me that the surface of the tire was rough, so the car would run slowly. He actually had a very initial understanding of the force of friction and he intuitively thought that the rough surface would make the car move slower, although he did not know why that happened."

Yet, children's intuitive thinking was not purely without foundation. As the teacher pointed out: "Although they have little basic knowledge about, or concept of, making a robot, their daily-life experiences may help them solve the problem. In the example I just mentioned, about the student removing the tire of the wheel, he told me that there were several little cars in his home and he often pushed them by hand. One day, he found that the car would sometimes move farther when the tire was removed. Therefore, he concluded that a rough surface might hinder the car's movement. That's how he came up that solution."

5.1.2 Development of Students' Problem-Solving Competencies: Qualitative Evidence

From the quantitative data, a significant increase in students' PS competencies was found from the pre-test (M_{pre} = 9.07, Mdn_{pre} = 9.21) to the post-test (M_{post} = 10.24, Mdn_{post} = 10.36), with z = -4.63, T = 48, and p < .001. In addition to these numeric data, this study found some evidence of the development of students' PS competencies in the qualitative data. The tool adopted for measuring students' PS competencies was based on the theoretical framework containing two domains: problem-solving self-efficacy (PSSE) and problem-solving skills (PSS). According to the definition in Chang et al. (2004), PSSE means "belie[f] in one's personal ability to solve problems successfully" (p. 15), while PSS refers to "the cognitive and behavioural activities by which a person attempts to understand problems and find effective solutions' or ways of coping with them" (p. 14). Thus, corresponding qualitative evidence of the development of students' PSSE and PSS will be demonstrated.

Problem-solving self-efficacy. From observation and interviews, some evidence showed that students' PS self-efficacy was enhanced. In the first week, because they lacked confidence in their work, when receiving negative signals from others, they might get confused or tend to copy classmates' work, or even give up on their work. Some students completed the program but did not immediately download to the robot; instead, they went to see their peers' work and compared others' programs to their own. If they found differences, they got confused, as they worried that they had made a wrong decision when programming. For example, in Lesson 1, the task was to write a Scratch program for controlling the motor, which was quite easy. One student (A06) completed his program independently, but he found that his program was different from others. His classmates also told that him his program was faulty. Therefore, he did not download the program until the teacher came to him:

Teacher: "You have finished programming. Why don't you download and have a try?"

Student: "Mine is different from others', so I'm thinking about how to revise my program."

Teacher: "How do you know your program is wrong and others' are right?"

Student: "But they are the same and they told me mine is wrong."

Teacher: "Okay, as far as I can see from your program, there is no obvious faulty part."

Student: "Really? But in their programs, these two parameters [of the motors] are different." (In this student's program, the parameters of the motors were all 100.)

Teacher: "No one has said that the two motors should run at different speeds, so both they and you are right."

From this case, we can see that the student was not confident enough, even though he did nothing wrong. Therefore, when he discovered that his idea was different from others' and when he received negative comments, he became hesitant. Other, similar cases could be found in the videotapes; some students directly copied other's works, whether or not they had finished their own programs. When the teacher discovered this and asked them for their reasons, their responses were like: "I don't know how to do it [programming]…" (B02), "I'm afraid that my program cannot be successful…" (B04), or "I thought his program was better than mine…" (B09). Essentially, they doubted their abilities to solve problems successfully, although they actually could do better than they thought. In the group interviews, when asked "How did you solve problems?," many students admitted that they copied others' work at first. Some explained:

Student A02: "I was not confident in myself, so copying would be the most convenient and easy way to succeed."

Student A06: "I knew copying was not right, but I thought my designs were worse than others."

Nevertheless, as the course went on, these phenomena were gradually changed. In the last week, although the lesson tasks were much more difficult than those of the first week, most of the students were more focused and passionate, did not give up immediately, and tried to solve problems independently. Some of them even tended to take on extra tasks. For example, in Lesson 11, the task was to make a robot follow people's steps, which is quite complex. Although there were some complaints among students at the beginning of the class, they immediately focused on solving the problem. When they had conflicting ideas, they talked together about them. The majority of students were able to complete the basic task independently, and several students could even explore and complete the extra task.

As the teacher said: "The confidence in problem-solving was established class-by-class [day-by-day]. They got used to the terrible feeling of failure and learned from it." When children realized that problems and failures are always there, they gained faith in their ability to successfully solve problems.

Interestingly, in the interviews with students, when they were asked, "Do you think you became more confident when solving problems? Can you provide examples?," students gave varied responses. Some thought that they became more confident than they were before:

Student D09: "Yes, I think so. When I made a robot before, I had no confidence because I was afraid of failure. But now, I feel better."

Student D08: "Yes, me too. I think I became more confident."

Researcher: "Do you have any examples?"

Student D08: "I can confidently teach other students when they are in trouble. I can share my experiences in building robots with my parents. I can describe [how to make a robot] in detail."

Student D09: "Oh, yes, I even shared my experiences with my friends [who are not in the summer camp] and I think I became stronger."

However, some thought that they did not become more confident. For example:

Student D01: "I'm not sure … because I think it [the learning content] became more and more difficult, and I still feel like I'm not able to learn robotics."

Researcher: "Do you have any examples?"

Student D01: "For example, when making the Dagao machine [Lesson 10], I almost gave up because the structure [of the machine] was very complex. There were many gears and other elements … I had to ask the teacher for help, and through the teacher's step-by-step guidance I finally finished building the machine. However, I did not feel good because I could not do it without the teacher's help."

From the examples above, we find that some students agreed that they were more confident than before, while some disagreed. These different voices might explain why there was a nonsignificant result in the measures of the students' perceptions of their own PSSE (M_{pre} = 4.81, M_{post} = 5.20, Mdn_{pre} = 4.86, Mdn_{post} = 5.29, z = -1.93, T = 326, p = .054 > .05).

Another interesting finding was that when students explained why they became more/less confident, resulting in more/fewer solutions, the chosen solution that proved to be successful/unsuccessful were mainly mentioned. For example:

Student A03: "Whenever I thought up new solutions, I felt so confident."

Student A06: "I felt confident because I succeeded in solving problems, especially when I saw my robot was successfully completing the tasks."

Student B06: "My robot did not move as I expected … it made me feel bad, so I was not confident when solving problems."

Student C02: "I was not so confident … because I failed to solve problems and I could not think of another solution."

This finding suggests that, in the stages of coming up with and evaluating solution(s), students' PS self-efficacy might be involved. Those who generated more solutions or inventive solutions and those who finally discovered that the chosen solution was effective might be more likely to believe in their own abilities to solve problems.

Problem-solving skills. According to the original theoretical model of measuring students' PS skills, there are four major skills: problem definition and formulation, generation of alternative solutions, decision making, and solution implementation and verification (Chang et al., 2004; Maydeu-Olivares & D'Zurilla, 1997). Next, some qualitative evidence of students' enhancements of each skill will be presented.

Problem definition and formulation. This skill was involved mainly in the stage of identifying the problem (Stage 1 of the problem-solving process). When identifying problems, a better problem solver must consider as much information about the problems as possible, including concrete facts, potential demands and obstacles, and realistic goals (Chang et al., 2004). In the classroom observation, the researcher noticed that, as the learning progressed, students' ability to define and formulate problems improved. In the early classes, students did not consider as many relevant facts as possible about the problems; however, gradually, they gathered more information to clarify and understand the problem.

In the first week of learning, as can be seen in their designs, students usually failed to collect relevant facts about the problem; thus, their designs might have been unreasonable. For example, in Lesson 2 (Make your vehicle 1 (Moving forward)), several students initially built a robot without wheels; some made a robot with wheels but the wheels were not connected to motors, while someone used four wheels of different sizes (such as two big and two small wheels), making the robot unbalanced. When the teacher coached these students, based on the talks between them, the researcher noticed that these students failed to understand some basic facts about making a vehicle, such as: a vehicle should have wheels; the wheels should be connected to motors that make the wheels turn; and the vehicle should be balanced when it is put on the ground. Therefore, they were unable to accurately formulate problems, such as: How to select suitable elements for making a vehicle? How to make the wheels turn? And, how to ensure the vehicle is balanced?

Through the accumulation of PS experiences, this situation gradually changed: Students asked for more relevant facts before exploring the problems and considered more potential problems, and the problems they formulated were more accurate. For example, in Lesson 6 (Make your vehicle 5 (An off-road vehicle)), some students asked the teacher about details regarding the off-road vehicle's structure, such as: What kinds of wheels are suitable for making an off-road vehicle? How can the caterpillar bands be settled on the wheels? And, how should the frame be built for settling the motors? Also, some other students carefully analysed the structure of the sample vehicle provided by the teacher, to understand its elements and mechanism. In Lesson 11 (A robot after you), when the teacher attempted to make students articulate their PS process, some students verbally described the problems they formulated. For example, when the teacher asked a student (D02), "What do you think is the problem you are going to solve in programming?," the student responded:

Student D02: "I think the main problem might be how to make the robot see [detect] the people in front of it and follow the people's steps."

Teacher: "Any specific problems you need to solve?"

Student D02 thought for several seconds and said: "More specific… How to program for the lighting sensors and for the motors? What blocks are suitable? What is the parameter that I should set for the sensors and motors? When should it [the robot] turn left and when should it turn right? When should it move forward and when should it move back?"

Teacher: "Great! Let's think about how to solve these problems…"

Although the classroom observation noted the students' changes, as in the examples mentioned above, some students did not agree that their ability to define problems had improved. In the interview, when asked, "Do you think your ability to define/formulate problems changed? Why? Can you provide examples?," a student (A05) responded, "I don't think so … I know what the problem was, but I don't know how to describe it." In addition, some thought that verbally formulating the problem was not necessary. For example, a student (C03) said:

"…The teacher asked me to describe the problem I'm going to solve and I told him [my thoughts], but he asked me to describe it more specifically. I was confused because I don't know why he wanted me to describe it in such a specific way. I don't think it's necessary."

However, we should note that the ability to verbally describe a problem is different from the skill of formulating problems, as it is possible that young children can think more than they speak. Therefore, when probing students' ability to define and formulate a problem, we should focus on not only how they state a problem, but also on how they consider relevant information about a problem, such as concrete facts, potential demands and obstacles, and problem-solving goals (Chang et al., 2004).

Generation of alternative solutions. This skill was involved mainly in the stage of coming up with solution(s), which requires the problem solver to generate as many solutions as possible (Chang et al., 2004). As learning progresses, students can generate two or more alternative solutions to a problem. For example, a common problem that students had to solve in various tasks was how to make a robot move in a straight line, as, due to being influenced by some forces, the robot might be unable to move in a straight line. Following is how students generated solutions to this problem in different lessons.

In Lesson 2, when making a moving-forward vehicle, many students noticed that their vehicles moved in a curved line, especially when they moved for a long distance. However, the students did not carefully consider this problem, because they were simply happy that the vehicle was moving. Therefore, no solution to this problem was generated in this lesson.

In Lesson 3, when making a move-and-return vehicle, students noticed that if the vehicle did not move in a straight line, it might be unable to return to its starting point. Therefore, they had to carefully consider and identify this problem. To solve the problem, some of them generated the solution of reducing the vehicle’s speed, as the vehicle would move in a straighter line when it moved slower. This solution worked, though it was not perfect: Sometimes the vehicle still moved in a curved line though, the angular deviation might have been smaller.

In Lesson 4, students were required to create a vehicle that could move forward and turn back when it touched the wall. In this task, the question of how to make a robot move in a straight line was a problem that had to be solved, because if the vehicle could move straight, the button of the touch sensor might not have been able to detect the wall. Some of the students still followed the old solution generated in Lesson 3 (i.e., making a slower vehicle). However, others generated a new solution: They observed their vehicles and discovered the rule of angular deviation (to the left or right). Then, when programming, they set different speed parameters for two motors to weaken the angular deviation. In addition, they thought up another new solution: Adding two driven wheels in front of two driving wheels. Thus, the vehicle moved straight.

In the last lesson, when the students created a simple football robot, the slow-down solution was no longer suitable, as the football robot had to find and kick the “football” quickly. Another solution—setting different speed parameters for two motors—was unstable when the robot chased the football. The solution of adding two driven wheels was also not desirable because this structure would find it difficult to make a turn. Therefore, some students generated a new solution: They used each motor to control two wheels on each side of the vehicle. The motors and wheels were connected by one 40-tooth gear and two 12-tooth gears. This new solution allowed the vehicle to move quickly in a straight line and turn quickly when searching for the football.’

Students generated not only more solutions but also inventive solutions. The generation of inventive solutions (i.e., original, creative, and effective ideas) was also considered evidence of PS skill (Barak & Zadok, 2009). In the last lesson, when solving the problem of how to make the robot move straight, as mentioned before, one student (D02) thought up a creative idea: “If there is a sensor that can detect the direction of the robot, then we can write a program to correct it immediately when it moves in a curved line and in the wrong direction.” Although this solution was not implemented because no such a sensor was provided in the robot kits, it would be effective in reality because there is a kind of sensor called a “gyro sensor” which has the function of detecting directions. Knowledge of the gyro sensor is beyond the students’ learning level, and this particular student had not previously acquired this knowledge. Nevertheless, she generated an inventive solution that was beyond her knowledge, which could be deemed evidence of PS skill development.

Decision making. “In decision making, the problem solver anticipates the consequences of the different solutions, judges and compares them, and then chooses the ‘best’ or potentially most effective solution” (Chang et al., 2004, p. 16). Decision-making skill was usually involved in the stage of making a decision. An example showing students’ decision-making process is as follows. For both Lessons 11 (*A robot after you*) and 12 (*A simple football robot*), a common problem to be solved was how to make use of the lighting sensor (i.e., a sensor that can detect an object in front of it by using a light transmitter; in Lesson 11, the ultrasonic sensor could also be used as the fungible). The object in Lesson 11 was people, while in Lesson 12 it was a lighting ball. Therefore, in Lesson 12, to solve the problem, some students adopted the same design as in Lesson 11. One student (B05) initially decided to create a same-structured robot but then changed his mind. When he started to build the robot, he suddenly stopped and smacked his forehead: “No! I should not make the same robot as before.” Other students wondered why. He said, “Because the position of the lighting sensor will be higher than the ball, which means it might be unable to ‘see’ the ball.” Consequently, he decided to set the lighting sensor lower than before and successfully solved the problem. From this example, we can see that this student anticipated the possible outcomes of the alternative solutions and finally made a decision to adopt the potentially most effective one.

Classroom observation revealed changes in the students’ decision-making skills. For example, as mentioned in the last section, students generated at least three solutions to the problem of how to make the robot move straight: a) reduce the speed; b) set different speeds for the two motors; and c) add two driven wheels. In fact, solutions b and c were more effective than solution a because even a vehicle at low speed still moved in a curved line, although the angular deviation might be smaller. However, during the first week, most of the students tended to choose solution a. As one student (C06) said, “It is easier to achieve, and I don’t need to change much on my programming and building.” Namely, in the early classes, students tended to choose the easier or more immediate solution. However, this situation gradually changed when they had to deal with the same problem in a more challenging context, such as football chasing. Though they had easier ways to solve the problem, most of them chose the most effective one—connecting each motor to two wheels on each side by gears. From observation, we can see that they were carefully considering the potential consequences when making decisions. For example, in Lesson 12, one student (C04), who chose the simplest solution (reduce speed) in the earlier classes, anticipated the possible consequence of all those alternative solutions and said:

"Setting a slow speed for my robot might solve this problem… But, according to my previous experience, the vehicle would still move in a curved line. I don't think it is suitable for a football robot. If I add [driven] wheels to it, it might be hard to make a turn, but the football robot often needs to seek the ball. A four-driving-wheel structure is not possible because I have only two motors. So, one motor should control two wheels of one side. Then the robot can move and turn fast."

The teacher also noticed that students changed their minds about making decisions, from choosing the easier solutions to choosing the most effective one:

"You can see their growth when the task became more difficult. They generated more solutions and were willing to try the most effective one, even though the most effective solution might be the most complicated one. They learned to think and do better than before."

Solution implementation and verification. A rational problem solver "carefully monitors and evaluates the outcome of the chosen solution after attempting to implement it" (Chang et al., 2004, p. 16) This skill was involved in the stages of carrying out and evaluating the solution and. In the early classes, students often monitored solution outcomes carelessly. For example, when solving the problem of how to make the robot move straight, as mentioned before, in Lesson 3 some students thought that reducing the speed would solve this problem. As a result, the outcome of implementing this solution was not desirable, because even at low speed, the vehicle still moved in a curved line. However, the students were satisfied with this solution and the corresponding outcome and, thus, did not come up with new solutions. When the teacher reminded them about this problem, some ignored him, while others said, “At least the vehicle is more likely to move a straight line.”

This situation changed when students carefully observed and evaluated the outcome of the chosen solution. In Lesson 12, when they encountered the same problem, they were not satisfied only with that undesirable solution (i.e., reducing speed). They not only generated more alternative solutions but also carefully judged and decided which solution might be the “best”, then evaluated the solution outcomes after implementation. For example, some students found that, through the addition of two driven wheels, the vehicle could move in a straight line. They did not simply accept this solution but carefully evaluated the solution outcome and found that the two driven wheels might lead to a new problem—it was hard for the vehicle to turn around. Therefore, they gave up this solution and turned to another one—using gears to connect each motor to two wheels on each side. Consequently, they found the new solution was suitable for completing the tasks of a football robot.

Moreover, from the teacher's observation, in the first two weeks, many students directly adopted or copied the solution generated by the student who first completed the lesson task, but they did not evaluate whether the solution was effective:

"When one student successfully completed the lesson task, other students stopped thinking and started to copy that student's work, especially the programming part. However, the worst thing was, they did not evaluate whether the other's program was the right answer and whether it was suitable for their own robots."

The teacher also observed the change in this situation in the last two weeks:

"During these two weeks, I noticed that students did not simply copy others' solutions. Although sometimes they had to refer to others' work, they also tried to carefully implement the solutions learned from others and evaluate the outcomes … I think they did learn something from others rather than simply copy others' work, because they could explain why the solutions were effective or not. This means they did evaluate those solutions on their own."

5.1.3 Effective Classroom Interactions in Students' Problem-Solving Development

Some evidence of the development of students' PS competencies was found in this research, as presented previously. Nevertheless, the improvement did not suddenly happen. In this research, the focus was not only on students' learning outcomes but also on effective classroom interactions that might contribute to students' learning outcome development. The correlation analysis identified a positive correlation between time spent on three interactions and PS competencies. However, the kinds of interactions that were effective in students' PS development should be probed in qualitative data.

Chang and colleageus' (2004) framework for PS orientations and styles categorizes problem solvers into two kinds: *constructive problem solvers* who have a positive orientation to problems and a rational style of solving them, and d*ysfunctional problem solvers* who have a negative orientation to problems and an impulsivity-carelessness or avoidance style of problem-solving. The pre-/post-test was built based on this framework, which means a higher score was more likely related to a constructive problem solver and a lower score was more likely related to a dysfunctional problem solver. From both quantitative and qualitative data, as many examples have shown before, most of the students at the summer camp were guided to be more constructive problem solvers. However, those who could not successfully be more constructive, or who even fell back to being more dysfunctional, also had to be carefully analysed.

Analysis of interactions from videotaped data identified four focal samples, including two whose measures of PS competencies were increased (closer to a constructive problem solver) and two that were decreased or slightly increased (closer to a dysfunctional problem solver). As shown in the Table 14, the pre-test scores of four individual focal cases were at a similar level (from 8.36 to 10.21). Next, four focal samples will be introduced. Through the tracking of their interactions with the teacher, peers, and robot, from the first class to the last one, effective and ineffective interactions that might explain their different development levels of PS competencies will be characterized.

Table 44

The Pre-/Post- Test Measures of Four Focal Samples

	No.	Gender	Grade	Pre-test	Post-test	Difference
Iris	D02	Female	2	8.357	10.429	2.071
Adam	D09	Male	2	9.000	10.785	1.786
	Average level			**9.071**	**10.243**	**1.172**
Bob	C06	Male	2	10.214	10.786	.571
Eve	C01	Female	2	9.071	8.429	-.643

Effective S-T interactions in students' problem-solving development. From the quantitative data, a significant positive correlation was found between the time spent on S-T interactions and the increase in students' PS skill (with *rs*(30) = .62, *p* < .001). In addition, from the qualitative data, effective S-T interactions promoting students' PS competencies were found.

In this research, a typical cognitive apprenticeship teaching model, which was deemed to be a useful model for improving students' PS skills at a robotics summer camp (Larkins et al., 2013), was used to guide the teaching process. Nevertheless, the way in which students interact with the teacher may lead to varied results. Therefore, for purposes of probing effective S-T interactions, both the cognitive apprenticeship teaching methods used at the summer camp and students' engagement interactions with the teacher will be presented.

Cognitive apprenticeship teaching methods. The cognitive apprenticeship model contains six teaching methods: modelling, coaching, scaffolding, articulation, reflection, and exploration. Findings from the observation and interviews showed that the teacher performed these six teaching methods adequately and properly in his teaching process, and that these methods could effectively enhance students' PS competencies.

modelling. This method requires an expert—such as the teacher, in this case—to demonstrate a task to the students and then allow them to observe and generate "a conceptual model of the processes that are required to accomplish it" (Collins et al., 1991, p. 13). Through externalization of the implicit processes, students can observe and understand the rationale for the PS process (Larkins et al., 2013).

Each class of the summer camp started with the teacher's demonstration of the lesson task and techniques that would be used to accomplish the task. The demonstration was intended to make implicit processes visible. For example, in Lesson 4, the task was to create a vehicle that could move forward and return if it touched a wall. The teacher drew the expected moving track of the vehicle on the blackboard, so that the students could understand the task goals. By showing the track on the blackboard, the teacher discussed the problems that had to be solved, such as how to make the vehicle move straight, how to make it return, how to detect the wall, and so forth. Also, the teacher showed a sample vehicle that he had prepared, and discussed the possible consequences of putting the touch sensor in different locations. When demonstrating the mechanism of a touch sensor, the teacher used the internal test function of the Intelligent Brick to show that: "As you can see, if we do nothing to the button of this touch sensor, the signal will remain a 0 value. However, when the button is pressed, you can see that it returns a 1 signal." By doing this, students could understand the two possible signs received by a touch sensor in a visible way: (button pressed – return "1") and (button released – return "0").

In this task, the most difficult part was programming, especially for the young students. Therefore, the teacher drew a flow chart following the algorithm on the blackboard and then used the sample vehicle to demonstrate the functions of different blocks in Scratch. For example, he showed the vehicle moving forward to the wall with the function of "repeat" in the Scratch program. When the vehicle touched the wall, it did not stop or return. Then he said, "If we use only the repeat block in Scratch, the vehicle will move repeatedly, so it will not stop or return. Therefore, we should add a condition for it. The If-Then block can set this condition. If the vehicle touches the wall, which means the touch sensor accepts a 1 signal, the motors should turn backward." He showed an example using the "If-Then" function; the vehicle successfully returned when it touched the wall. In addition, the teacher handled the vehicle to simulate this process and said, "As you can see, the vehicle goes forward … Okay, now it touches the wall and the button of the sensor was pressed, so the robot accepts a ‘1’ signal. Then the motors are turning backward … Now it comes back to the starting point. This is the task you are going to accomplish today.” In this case, by demonstrating the task, the techniques involved in this task, and possible problems that might occur in visible ways (i.e., drawing on the blackboard, using a sample vehicle, putting hands on the vehicle), the teacher helped the students understand the PS process and construct a conceptual model that could be adapted into their own work.

Additionally, the modelling activities may be especially effective in developing the skill of *Problem definition and formulation*. For example, in Lesson 3, a student (D03) found that her vehicle did not move and asked the teacher for help. The teacher demonstrated his own PS process to this student: “Firstly, we should check the battery of this Intelligent Brick.” Meanwhile, he turned on the Intelligent Brick and showed the battery icon to the student: “You can see that the electricity is sufficient. Then we should consider whether the connection of the motors is working.” He opened the internal test interface and selected “motors - forward.” The motors were running normally, so he said, “You can see that the motors are connected correctly. Then the problem might be with the programming.” He started to check the student’s program. “As we can see from your program, you put only one If-Then block here. Let’s imagine that when the robot reads this block, it will judge whether the touch sensor received a 1 signal. If YES, it will stop and return. If NO, the robot will skip to the next step. However, you did not write a next step for it, so the robot was waiting for a 1 signal without any movements. This is the problem. This is why the robot did not move." The teacher then added another If-Then block and said, "By adding this block, now, we tell the robot, if the touch sensor receives a 0 signal, it should run forward. Without this function, the robot will not move and wait until the touch sensor is pressed." In this case, the teacher described his thinking process in verifying a problem, which allowed the students to understand the reasoning and rationales behind the problem definition. Modelling is especially helpful in identifying the problem stage, compared to other stages of the PS process, because identifying the problem is challenging for young students (Eguchi, 2012). Therefore, an expert's modelling demonstration is necessary.

coaching. Coaching involves observing and assisting students while they perform a task, such as providing "hints, scaffolding, feedback, modelling, reminders, and new tasks aimed at bringing their performance closer to expert performance" (Collins et al., 1991, p. 14). This usually involved students raising specific problems or at critical moments (Collins et al., 1991; Larkins et al., 2013).

In each class of the summer camp, after the teacher's modelling demonstration, students started carrying out the task. In this process, the teacher carefully observed individuals' performances. When someone was trapped by a problem, the teacher stopped by the student and observed the student's behaviour. "I usually observed someone who had difficulties with his/her work, for around one or two minute(s). If he/she was truly unable to solve the problem, I thought it was time for me to help." From observation, when a student (D04) used an inappropriate element for building his robot (e.g., he chose a connector peg with *friction* to connect a wheel to the beam, while the proper way was to use a connector peg with the *axle*), the teacher coached him: "You can see that the connector you have has two rounded heads, but your wheel has a cruciform hole. That is why this connector cannot be inserted into the hole. Do we have another, more suitable connector?" Consequently, the student found a connector peg with a cruciform axle.

The coaching method, for students, was especially effective in terms of students' skill development in the area of *Generation of alternative solutions*. Because young students often used intuition to generate solutions, as stated previously, the teacher's coaching can help them cognitively understand the tacit process of generating a solution. In the example (D04) mentioned above, the teacher showed his thinking process of generating a solution. He compared the shaping of two elements (i.e., the *rounded* head of the connector and the *cruciform* hole of the wheel) and reminded the student of another proper choice for the cruciform hole. This coaching activity allowed the student to understand the thinking process behind generating a solution: 1) When two elements are not matched, it might be because of different shapes; and 2) To solve this kind of problem, observing and considering the shaping of the elements and searching for another matched element might be a solution. As a result, the students' PS skills were cultivated.

scaffolding. To scaffold is to support students by providing suggestions, help, or even physical support (such as cue cards or robot kits), especially when they encounter difficulties requiring skills beyond the students' abilities (Collins et al., 1991; Larkins et al., 2013). In a robotics learning environment, there are many resources for scaffolding, such as the robot kits, the internal-test model of Intelligent Brick, the sample programs of Scratch, and so forth. During the summer camp in this research, the teacher sufficiently utilized those resources for scaffolding students. For example, in Lesson 9, the task was to create a walking robot. The structure of a walking robot is complex, which meant many students had to solve several problems when building it, though the teacher had already demonstrated the structure of a walking robot at the beginning of the class. Therefore, the teacher prepared a sample robot for the students. When they sought help, the teacher used the sample robot to help them clarify problems and demonstrated, in detail, the mechanism of different parts of the robot.

The timing of scaffolding is important. The teacher should diagnose the level of students' current skills and provide appropriate assistance for students to achieve higher-level skills (Collins et al., 1991). For example, in this research, the teacher offered different levels of support to students at different levels. As the teacher stated:

"For those whose problem-solving abilities were stronger than others, I just gave them some hints; but for those who were relatively weaker, I provided physical support with more details."

In addition, “as the student step-by-step performs better, the teacher gradually fades his or her helping process” (Järvelä, 1999, p. 253). During the first two weeks, when many students were unable to solve problems independently, the teacher provided more support to them. The classroom observation revealed that he was quite busy with scaffolding students. However, he was increasingly “leisurely” during the last two weeks, when most of the students were able to work independently.

The scaffolding method was especially useful in developing students’ skill of *Generation of alternative solutions*. For example, in Lesson 6, the teacher drew a flow chart to help students understand the programming process behind making the vehicle turn right/left. In the flow chart, the teacher wrote down that the vehicle moves forward for 10 seconds, then turns right for 3 seconds, and then moves forward for 10 seconds. However, when students wrote their own programs, they not only adopted the teacher’s solution shown in the flow chart but also copied the parameters for time controlling (i.e., 10 seconds and 3 seconds). As a result, the students might have found that their vehicles did not turn in a 90° angle because the “3 seconds” was set in particular for the teacher’s sample vehicle. For different vehicles, this parameter should be correspondingly adapted. Students simply followed the teacher’s solution without logically understanding the meaning behind it. In the interview, the teacher mentioned this phenomenon and explained his scaffolding process: “At that time, I knew they did not truly understand programming but only copied what I drew on the blackboard. Therefore, I guided them in using the internal test model of the Intelligent

Brick and observing the different angles in which the vehicle turns by setting different parameters for it. By doing this, the students can see that setting a parameter in programming control the duration of movement. You need to find a proper value to make the vehicle turn in a 90° angle rather than just copying my '3 seconds.'" In this case, after the teacher's scaffolding, students tried to find the proper solution for their own vehicles. Some even tried to make their vehicles turn 180° or other angles, which indicated that their ability to generate solutions was enhanced.

articulation. Articulation is to guide students in making their tacit knowledge, internal reasoning, or PS process concrete and explicit (Collins et al., 1991; Järvelä, 1999; Larkins et al., 2013). Leading young-aged students to articulate their internal thinking is hard because their ability to orally present is at a limited level and their thinking process might be illogical. Nevertheless, in this research, by using the inquiry teaching strategy, the teacher encouraged students to think aloud as best as they could. For example, when a student (B06) was solving the problem of how to make the robot move straight, the teacher asked her the following questions:

Teacher: "From your observation and experiences, in what condition does the vehicle move in a curved line?"

Student B06: "I found that when it moves for a long distance, it deviated more and more from a straight-line track."

Teacher: "Do you think moving for a long distance might be a reason for this situation [moving in a curved line]?"

Student B06: "I guess not."

Teacher: "Why do you think so?"

Student B06: "Because the long distance might make this situation more obvious, but in the short distance, this situation was so slight that we did not notice."

Teacher: "So, what might be a reason for this situation?"

The student fell into silence for several seconds.

Teacher: "Let's rethink about the condition in which the vehicle moves in a curved line. Do you have another finding?"

Student B06: "Oh, I remember that your vehicle [the sample] had four wheels and it moved straight."

Teacher: "So, how many wheels do you have?"

Student B06: "I also have four wheels … But they are different!"

Teacher: "What's the difference?"

Student B06: "Your four wheels are all the same size while I have two big wheels and two small ones."

Teacher: "Any other differences?"

Student B06: "You put two wheels on the left side and two on the right side. My vehicle has two big wheels on the middle part of the right and left sides and two small wheels on the front and back sides."

Teacher: "Do you think that might be a reason why the vehicle moves in a curved line?"

Student B06: "Yes, because your vehicle seems more balanced than mine."

Teacher: "So, you think your vehicle is not balanced enough; why?"

Student B06: "Maybe because of the different structure and location of the wheels."

Teacher: "So, now, what is your decision?"

Student B06: "I will refine my vehicle's structure."

As we can see from this case, through systematic questioning, the teacher encouraged the student to analyse a problem and guided her in articulating the thoughts of generating solutions and making decisions. Although in the end she could not explain the principles behind the solution, she had the opportunity to think aloud and externalize her internal thinking process.

Articulation might be especially beneficial in the skill development of *Generation of alternative solutions* and *Decision making*. Also, students' PS self-efficacy might be enhanced by articulation because they have the opportunity to express their thought process. For example, in the interview, the student (B06) in the case above said, "The teacher did not simply tell me what is right and what is wrong. He encouraged us to think and rethink and patiently talked with us. He let me feel that I have the ability to solve problems."

reflection. Reflection allowed students to compare their own performance in the problem-solving and thinking process to that of the expert (Järvelä, 1999). In the last stage of every class during the summer camp, the teacher organized students in reflecting on the problems they encountered and solved when carrying out the task. Then they displayed their productions one by one. The teacher also showed his own sample robot and presented his thinking process when solving problems. For example, when showing a sample football robot, the teacher said:

"This is my football robot. As you can see, it contains two lighting sensors located on the front and back. I used two rather than one lighting sensor because it reduced the amount of time consumed when searching the football. Of course, you can use more than two lighting sensors to quickly find the football, but you should consider the cost and programming problems as well. To quickly find the football, at the same time, the robot should turn as fast as it can. Therefore, I used two big wheels connected to motors as the driving wheels and two rotary wheels as the driven wheels. As you can see, it turns very fast because the force of friction is relatively low. However, as you have found, I also faced the problem of the robot not moving in a straight line. I solved this problem by refining my programming. [The teacher displayed his program to the students.] You can see that when the robot is going

forward to a football, by the 'If-Then' and 'Repeat' function in the program, it is still judging whether the football is in front of it. If the football is lost, the robot can turn around and search for the right position of the football."

After the teacher's presentation, students commented on their own work by comparing it to the expert's (the teacher's) work. For example, a student reflected on his robot:

"My football robot has only one sensor on the front side, while the teacher's robot has two sensors. It is as though you have other eyes on your back, so you can quickly find the subject. Additionally, I used a four-driven-wheel structure because I thought it could turn quickly and also move in a straight line. But the teacher's robot turns faster than mine does and it can go forward to the football accurately because the teacher wrote a better program for it. I did not think of how I could refine it in programming."

Because the reflection activity was for students to review the whole PS process, this method was useful for promoting all PS skills. However, requiring students to compare their works to others' might hinder students' self-efficacy development. As one student (C06) said, "It made me feel anxious when others' work was better than mine." Therefore, it is important to tell students that reflection is not for comparing but for reviewing.

exploration. Exploration is to push students into a less-structured environment to probe tasks of their own interest (Collins et al., 1991; Larkins et al., 2013). In the first two weeks of the summer camp, students followed a more prescribed program, which was intended to build and consolidate their foundational knowledge of robotics and the basic skills of PS. From the third week, the teacher encouraged students to explore more possibilities beyond simply completing the settled task.

For example, in Lesson 7, the task was to create a fan that could be turned on and off by pressing a button. Because the students were previously equipped with basic knowledge about the touch sensor, most of them could easily complete this task. Therefore, the teacher set a new goal for them: to make a fan with more functions. This goal was general, without specific guidelines, so the students were allowed to explore more possibilities regarding making a fan. Some students made a fan with different turning speeds controlled by the touch sensor. Some used a lighting sensor to create a fan that could automatically turn on when someone was in front of it. Some made a fan that could use motors to horizontally turn around.

During the exploration process, students solved the problems that interested them. Because students' exploration might involve every stage of the PS process, the exploration method could be helpful in developing all PS skills.

Students' engagement in cognitive apprenticeship teaching. The cognitive apprenticeship teaching was deemed to be a useful technique for developing students' PS competencies (see, for example, Larkins et al. (2013) in the robot education domain); this research also found that cognitive apprenticeship was effective in developing PS competencies. Nevertheless, students' learning outcomes varied. Therefore, "it would be useful to characterize exactly which students benefit and which do not benefit from a certain method" (Järvelä, 1999, p.261). Namely, much crucial information about students' intra-class variations in S-T interactions should be provided (Good & Brophy, 1971).

In this research, through the characterization of those who were becoming more constructive and those who were becoming more dysfunctional, the findings showed that the question of whether students were actively engaged in S-T interaction was significant to their PS competencies.

active engagement. Iris is a grade-2 girl whose PS competency measure increased by 2.07 (M = 1.79) from the pre-test to the post-test. Namely, she became increasingly close to being a constructive problem solver. By tracking her performance during the summer camp, Iris showed active engagement in S-T interaction. For example, she always dedicatedly followed the teacher's modelling. When the teacher was coaching and scaffolding her, Iris actively talked with the teacher rather than simply responding to the teacher. The teacher's articulation process was smooth in her case because Iris was willing to express her thinking.

Moreover, she actively reflected on her own works and that of others. In some exploration tasks, she tended to challenge the more difficult ones. During the first two weeks, her problem-solving orientation was positive, while PS style was with impulsivity and carelessness. Meanwhile, she sufficiently utilized the support and resources provided by the teacher, and thereby, in the last two weeks, performed better and started becoming a constructive problem solver with a more independent and rational PS style.

inactive engagement. Eve is a grade-2 girl whose PS competency measure decreased by .64 from the pre-test to the post-test (*M* = 1.17). Namely, she was getting further away from being a constructive problem solver. By tracking her performance in the summer camp, Eve showed inactive engagement in S-T interaction, unlike Iris. For example, when the teacher demonstrated the task, Eve played with the elements of the robot kit herself. She did not carefully follow the teacher's modelling. To the teacher's inquiries, she responded carelessly or did not respond at all. When the teacher was coaching and scaffolding her, Even simply listened and followed the teacher's instruction, without engaging in in-depth communication with the teacher. The teacher's articulation process was tough in her case because her responses were unmindful. Moreover, Eve rarely reflected on her own work or that of others. She was satisfied with completing the initial-level task and showed less interest in exploration tasks. During the first two weeks, she solved problems mainly by copying others' ideas, such as the robot structure and program. Often, she learned something from copying others' work. However, during the last two weeks, she still could not complete her work independently. Although the teacher attempted to stimulate her to articulate her thought process and offered many suggestions and supports to her, Eve simply got lost in playing with the elements. Sometimes she even gave up and chatted with others when problems arose.

Another focal sample, Bob, is a grade-2 boy whose PS competency measure slightly increased by .57 from the pre-test to the post-test (*M* = 1.17). He also showed inactive engagement in S-T interactions. For example, in Lesson 7, Bob made a simple fan without a motor; therefore, the fan did not turn. When the teacher tried to guide him to add a motor, Bob listened and nodded, but when the teacher went forward to another student, Bob did not take action to refine his work. Instead, he played with the fan alone. When a classmate asked him why his fan did not have a motor, he said, "I can turn it with my finger."

Reflecting on the examples above, and other, similar examples at the summer camp, the researcher found that successful cognitive apprenticeship teaching relies on students' active engagement in S-T interaction, as S-T interaction involves both teacher and student participants. Therefore, the findings showed that those whose PS competency measure was significantly enhanced shared a common characteristic in that they were actively engaged in S-T interactions, while those who did not perform better in PS showed the opposite characteristic.

Effective S-S interactions in students' problem-solving development. Previous studies suggested that group collaborative interactions might be helpful in promoting students' PS development (Collins et al., 1991; Järvelä, 1999; Larkins et al., 2013; Mitnik, Recabarren, Nussbaum, & Soto, 2009); however, in this research, the teacher did not organize group activities for several reasons:

"Unlike a normal learning environment in school, the robotics summer camp is more open and has a lot of room for students to interact with others. If we intentionally group them or pair them, their interactions might be limited to groupmates or partners. Besides, every child wants to have their own robot, while in group work, there might be only one product. Children's sense of achievement usually stems from making a special robot that belongs only to himself/herself. Additionally, building a robot is a kind of hands-on work; it is hard for more than one person to put their hands on a robot. Those who cannot directly work on the robot in group work might feel frustrated and unsatisfied, as every child wants to play with the robot's elements."

The researcher of this study agreed with the teacher's rationales and respected the teacher's decisions because his considerations were more matched with young children's demands, which were aimed at providing pupils with a healthy learning environment. There is no reason to force students to work in small groups or pairs and to ignore their natural instincts of creating their own robots. Additionally, in a previous study of Yuen et al. (2014), students working in a group had to wait for their turn to use robot kits or computers; therefore, children who were waiting could also have been doing off-task things such as chatting and playing with others. Considering the reasons above, this study allowed children to work individually on their own robots and computers, though interactions between or among students were also encouraged. Also, this arrangement will be marked as a limitation of this study; future studies might focus more on the balance of students' demands and effective group activities.

It should be noted that, although no group/pair-formed activity was involved in the summer camp, S-S interactions still existed. This is because the class size was small (fewer than 10 students) and all students were gathered as one "group" which allowed them to communicate and interact with each other freely. As shown in the last chapter, the time spent on S-S accounted for 41% of the duration of the summer camp, which was more than the other two kinds of interactions (i.e., S-T and S-R interactions). It also aligns with the teacher's statement that "the robotics summer camp is more open and has much more room for students to interact with other each other." In addition, from the quantitative data presented previously, there was a significant positive correlation between the time spent on S-S interactions and the increase in problem-solving competencies (*rs*(30) = .63, *p* < .001). The correlation was especially strong in students' PSSE increase (*rs*(30) = .55, *p* = .001). Therefore, we should focus on effective S-S interactions in the development of PS competencies when analysing qualitative data, though there was no group/paired-formed activity at the summer camp.

In the original conceptual framework of this research, S-S interactions contain two kinds of cooperative interactions (i.e., discussion and observational learning) and off-task interactions. By tracking four focal samples' performances from the first class to the end of the summer camp, this research found that cooperative interactions were effective, while off-task interactions (such as unstructured chatting, playing with the robot or peers, and other interactions unrelated to the lesson task) were ineffective for the development of students' PS competencies. Specifically, in addition to discussion and observational learning contained in the original framework, the division of work was also included in the effective S-S interactions according to the qualitative analysis. Therefore, three effective cooperative interactions found in this research are discussion, division of work, and observational learning.

Discussion. S-S discussion involved idea-sharing and conflicting. Through idea-sharing and conflicting, students exchanged different ideas to solve shared problems. For example, when determining how to make the robot move straight, which was a common problem in robotics, a focal sample was how Iris shared her ideas with others: "I think we should consider four wheels with tires, just like the car we usually see in daily life." Another student (D04) then raised his concern: "But we have only two motors." This concern aroused another student's (D07) idea-sharing: "We can use gears to connect them." Through these meaningful discussions, the students thought up alternative solutions to the problem. Also, because different ideas were communicated, their understanding of the problem was getting deeper. In the interview, many students stated that they learned from peer discussion. For example:

Student A03: "I usually talked with my classmates when I was thinking about how to solve the problem, and sometimes, they provided some useful ideas."

Student B02: "If there was a problem that I could not figure out, I discussed it with other classmates firstly … because I think they know more than I do."

Student C04: "…discussing with them [peers] was helpful because they usually came up with ideas that I did not think of."

Discussion also allowed students to think aloud. Sometimes children may prefer to discuss things with peers rather than with the teacher. From classroom observation, the researcher found that peer discussion was a good supplement to the teacher's articulation. Those who talked less to the teacher may have discussed more with peers, so they could articulate their PS thinking process to peers, which might also contribute to their PS skills. Moreover, they usually talked with peers when reflecting on others' work. They often shared reflections with peers when someone else displayed his/her robot. Sometimes they were shy about directly giving feedback to a classmate they were not familiar with, but they would discuss the matter with close peers in private. We should critically treat this kind of private discussion. On the one hand, those who were expecting feedback may lose the chance to hear from others, while on the other hand, those who had discussions with peers may benefit from idea-sharing and conflicting.

Division of work. In qualitative data analysis, division of work was added to the conceptual framework, because surprisingly, although students were young and they had to deal with relatively complex tasks, sometimes some children showed the idea of division of work. For example, when searching for elements, which was an aspect of robot-building that consumed much time, the focal sample Iris said: “We should separately find elements.” Then she led her classmates to find different elements, so that they quickly found all the key elements for building the robot. When solving the problem of how to make the robot move straight, another focal sample, Adam, said that: “We can try different plans separately to see which one is better.” By arranging for other classmates to divisionally try different plans, they did not have to try as many times to find the better one.

Division of work was deemed an effective S-S interaction for developing students’ PS competencies because, on the one hand, the “leaders” (such as Iris and Adam) became more confident when leading classmates to solve problems. As Iris said, “…it made me feel confident when they agreed with me and listened to my arrangement.” On the other hand, other students participating in the work division knew that there were peers they could rely on, so they also felt confident when solving problems. In addition, work division was time-saving, so students had more time to work out other problems and accomplish those tasks, which would also enhance their PS self-efficacy and skills.

Observational learning. Students' observation was directed toward gathering information from peers and transferring information back to their own work. Commonly, in one classroom, someone might solve problems more quickly than others. Therefore, he/she became an "expert" among peers; other students could observe how the expert solved problems and could learn from the expert's experiences. The two focal samples Iris and Adam took advantage of observing peers' PS process and gradually became the experts. For example, in Lesson 3, when making a vehicle (move forward and return), Iris observed how a classmate measured the turning time of a robot; then she adopted the classmate's method to measure her own robot's turning time. Afterward, in Lesson 4, she applied the same method learned before. In Lesson 6, because the task was different from the previous one, she adapted the method and successfully applied it to the new task; therefore, some other students observed her work.

In addition to observing peers' work, students learned from observing S-T interactions. For example, when the teacher was coaching and scaffolding a student, Adam observed their communications and gained useful information for his own work. The videotaped data showed that when the teacher was coaching a student in computing (i.e., programming and debugging), Adam came closer and watched/listened to them. Sometimes he got important information that was helpful for his problem, so he showed an excited face and went back to his work. Other times, he was still confused so he joined the S-T interactions and discussed his problems.

Through observational learning, students learned the lessons of others' success and failure when solving problems; therefore, their PS skills were enhanced. However, some counterexamples rarely participated in observational learning, and they did not benefit from such kinds of S-S interactions. For example, the focal sample Eve showed less interest in observing others' performances. She preferred to focus on her own work; this was not a bad thing, but even when she was unable to solve the problem, she did not try to observe and learn from others' experiences. Therefore, her work progress was usually slower than others and most of the questions she asked the teacher were those that others had solved. Hence, as presented previously (in section), she disagreed that she became more confident when solving problems and even thought that "I'm not suitable for learning robotics."

Besides the three kinds of effective S-S interactions above, students' off-task interactions, such as unstructured chatting, playing with the robot or peers, and other interactions unrelated to the lesson task, were included in the original conceptual framework. However, based on the analysis of qualitative data, the findings did not support the idea that off-task interactions were effective for developing students' PS competencies, as students might get lost in off-task interactions. For example, Bob often chatted with others about topics unrelated to the lesson task, so his working progress got slower. Also, for those who participated in Bob's chatting, their work progress was also interfered with because they could not focus on their work. In addition to unstructured chatting, playing with the robot or peers might hinder students' learning. Classroom observation revealed that, most of the time, Bob was able to build a robot with the basic structure, but after completing the building task, he did not go to the program for the robot; instead, he downloaded the preset program that was unrelated to the lesson task and played with the robot. Consequently, he got lost in playing with the robot and forgot to complete the lesson task unless the teacher reminded him.

In summary, discussion, the division of work, and observational learning were found to be effective S-S interactions that might be helpful for developing students' PS competencies, while off-task interaction was not included in effective S-S interactions. What should be noted is that, although quantitative results indicated a nonsignificant correlation between students' time spent on S-S interactions and their PS skills, some qualitative evidence indicated that participating in peer discussion and observational learning might enhance not only students' PS self-efficacy but also their PS skills. In addition, the reason why work division was added to effective S-S interactions was mainly because it might lift students' PS self-efficacy, as discussed before.

Effective S-R interactions in students' problem-solving development. From the pre-/post-test data, students' enhancement of problem-solving competencies was significantly positively correlated with time spent on S-R interactions (*rs*(30) = .46, *p* = .008). From the qualitative data, this relationship and the important role of S-R interactions are also found. For example, Iris (the focal sample who got greater enhancement of problem-solving competencies) said in the interview:

Researcher: "Do you think there were interactions between you and your robot?"

Iris*: "Yes. I often interacted with my robot. For example, I ordered it to move forward and turn back, and then it would do what I told it to do."*

Researcher: "What do you mean by told it to do'?"

Iris*: "I mean I programmed for it. I wrote a program to control it."*

Many other interviewees shared this kind of interaction. In addition to the programming part, when they were building a robot, there were some S-R interactions, as the teacher said:

"When building a robot, it is inevitable to interact with that robot. For example, when settling a touch sensor on a robot, you have to try to find out whether the sensor can possibly touch the object. If you put it too deep inside the [body of] the robot, the sensor might hardly be able to touch the object. That is why you can see that sometimes students would use their hands to move their robots to hit the wall. Some students may find that the wheels touched the wall first, so the sensor could not [touch the wall]; consequently, they would refine the

structure of the robot. I think this is a kind of interaction, but I have to say that the interactions between student and robot were mainly on the programming part."

In the original conceptual framework of this research, building, programming, observing, and participatory investigation were included in S-R interactions. However, during analysis of the qualitative data, it was found that structure refining and program debugging were more suitable for characterizing effective S-R interactions. As discussed before, two typical kinds of problems involved in robotics learning were building problems and programming problems; thus, when analysing S-R interactions, it is reasonable to see how students solve these two kinds of problems by interacting with the robot. Also, compared to "building" and "programming" which describe students' one-way behaviour, "structure refining" and "program debugging" are more suitable for describing interactions. Consequently, structure refining and program debugging were characterized as two kinds of effective S-R interactions in developing students' problem-solving competencies.

Structure refining. This kind of interaction was for solving building problems. When building a robot, students were implementing their design ideas; however, there was often a gap between their ideas and the handmakes. This is where building problems may come from. Structure refining is not only a common method for solving building problems but also a kind of S-R interaction that involves all PS skills. For example, when making a vehicle in Lessons

2 and 3, some students settled the Intelligent Brick under a frame and connected the wheels directly to the holes of the frame. It was an idealized design that seemed like “no problem”, as Adam said: “One frame with four wheels. This is what a common car looks like.” However, when Adam started the vehicle, it moved slowly and not smoothly. After observing this situation for a while, Adam said, “The vehicle is hardly moving. Just like it was hindered by something.” Therefore, he stopped the vehicle and checked his work thoroughly: “It seems like no problem…” Then he used his hands to turn the wheels and, this time, found the problem (The skill involved: Problem definition and formulation): “Oh, I know. The wheels were too close to the frame, so the wheels were rubbing the frame. That is why it hardly moves.” He took the wheels and connectors out of the frame and tried to find alternative elements “for separating the wheels and the frame” (The skill involved: Generation of alternative solutions). Through trying different elements, he decided to add bushings for separating the wheels and the frame (The skill involved: Decision making). Consequently, he tried again to run the vehicle and observe the effectiveness of the new solution (The skill involved: Solution implementation and verification). Finally, after he fine-tuned some of the details, the problem was successfully solved. In this focal case, Adam’s structure refining involved all the skills of PS. By repetitive interactions with the robot, structure refining allowed students to practice PS skills. Also, through the structure refining process, the PS process was handed on. Moreover, students gained more confidence from this process.

On the other hand, Eve showed fewer refining behaviours, even when she saw that a problem occurred. For example, when she found that her vehicle did not move straight, she ignored it. Her reason was "it is not a big problem." Therefore, the interaction between Eve and her robot was cut off. She ignored meaningful information from the robot's behaviour, so, in the end, she did not refine the structure of the robot and could not benefit from the problem-solving process.

Program debugging. This kind of interaction was for solving programming problems. Debugging is an inevitable part of programming, for finding and resolving problems in a program. For example, in Lesson 4, Iris wrote a program for her vehicle and expected that the vehicle will move forward and return when it touched the wall. However, she found that the vehicle did not return. Therefore, she picked up the vehicle in her hands and ran the program again. She observed the wheels' rotation. Both wheels turned clockwise (moving forward) when the vehicle was started. Then she pressed the button of the touch sensor and the wheels stopped turning. By doing this, she realized that the problem was with the "If-Then" block setting for controlling the touch sensor and motors. Then she checked her program on Scratch and found that: "No wonder it stopped. I set that if the touch sensor was touched, then the motors' rotate speeds would turn to zero. I should modify them to 100" (The skills involved: Problem definition and formulation, Generation of alternative solutions, Decision making). Then she modified the parameter of two motors under the "If-Then" block and ran the program again; however, this time she found that the vehicle still did not return (The skill involved: Solution implementation and verification). She observed the vehicle again in her hands and found that when the button of the touch sensor was touched, the wheels continued turning clockwise. Therefore, she knew that: “Oh, the rotation direction is wrong. I forgot to set the motors to turn anticlockwise” (The skills involved: Problem definition and formulation, Generation of alternative solutions, Decision making). Consequently, she modified the motors’ rotation direction under the “If-Then” block and, finally, the problem was successfully solved. All the skills of problem-solving were involved.

On the other hand, for the example of Bob, his programming process was usually one-off. He wrote a program and downloaded it to the robot. A few times, his program as successful without debugging, but often, when problems occurred, he did not carefully analyse the robot’s behaviours and check his program. Instead, he simply sought the teacher for help or even tried to copy other students’ programs.

5.1.4 Summary

The findings showed that there were three features of problem-solving. First, problems that should be solved in robotics learning environments were mainly of two forms: building problems and programming problems. Second, children’s PS process generally followed a five-stage cycle: identify the problem, come up with a solution, make a decision, carry out the solution, and evaluate the solution. Third, children's solution-generation process was usually intuitive.

Also, qualitative evidence showing students' development of PS competencies, including both self-efficacy of PS and PS skills, was found.

Moreover, it was found that three kinds of interactions (i.e., S-T, S-S, and S-R interactions) played important roles in the development of students' problem-solving competencies. How these interactions influenced students' PS competencies was presented. In S-T interactions, the teacher's cognitive apprenticeship teaching methods and students' active engagement contribute to students' problem-solving competencies. With cognitive apprenticeship teaching methods, students could think aloud, which benefitted the students' PS competencies if the students actively engaged in the teacher's teaching process. Although no group/pair-formed activity was involved in the summer camp, S-S interactions still existed. Among S-S interactions, discussion, the division of work, and observational learning may be helpful for students' PS competencies, because: a) Discussion was a good supplement to the teacher's articulation when the student preferred to interact with peers rather than the teacher; b) Division of work was effective and time-saving, so students had more time to work out other problems and implement the task, which would also enhance their PS self-efficacy and skills; and c) Observational learning allowed students to learn from an "expert", which may have led them to become another expert. As for S-R interactions, structure refining and program debugging were characterized as two kinds of S-R interactions that may influence students' problem-solving competencies. Structure refining is for solving building problems and program debugging is for programming problems; both interactions involved all PS skills.

In addition to the findings above, we should note that three student-related interactions often overlapped. When a teacher was scaffolding a student, the student may also interact with a robot. During interactions with a robot, S-S observational learning may also occur. Also, these interactions often worked together to contribute to students' PS competencies. For example, when students were discussing how to solve a problem, interacting with a robot may have effectively promoted the discussion.

5.2 Students' Learning Outcome: Computational Thinking Skill

In this chapter, based on the qualitative data, three crucial pieces of information about students' computational thinking skills will be presented. The first is the feature of computational thinking, which includes the following: a) it focuses on programming problems, b) it follows a "three As" circled process, and c) it is related to problem-solving competencies. Also, some evidence of students' computational thinking skill development was found. Moreover, the role of three interactions (i.e., S-S, S-T, and S-R interactions) in students' CT development will be presented.

5.2.1 Features of Computational Thinking in Robotics Learning

Focusing on solving programming problems. Problem-solving competencies were discussed when students solved both building and programming problems at the robotics summer camp, while computational thinking skill was for solving programming problems. Specifically, at the summer camp of this research, which was for young-aged children, programming problems usually stemmed from three typical programming logics: *conditional logic* (such as If-Then block in Scratch), *iterative logic* (such as Repeat/Forever blocks in Scratch), and *parallel logic* (such as multiple If-Then). These programming logics were also deemed to be component elements of computational thinking in early literature (such as in Maloney, Peppler, Kafai, Resnick, & Rusk (2008)). Helping students understand the differences among these logics and properly adopt these logics is the main task of teaching CT skills. Next, several examples from the observation data will be presented to elaborate on the programming problems stemming from these programming logics.

Problems in conditional logic. In Scratch, “If-Then,” “If-Then-Else,” “Wait-Until,” and “Repeat-Until” blocks are available for achieving conditional logic. However, due to the limited comprehension of conditional logic, students might have encountered some problems when applying these blocks. Sometimes, they got confused with these blocks. For example, when ordering a robot to move and return if it touched the wall, some students adopted the “Repeat-Until” block to make the robot move forward until it touched the wall. However, their robots touched the wall and did not return because, when the conditional expression setting for “Until” was true, the robot would quit repeating (moving). This kind of problem stemmed from the fact that they rarely understood the conditional logic behind expressions.

Another typical problem involved setting a condition. In the “If-Then” block, several common conditional expressions could be used for setting a condition; for example: greater than (with a “>” sign), less than (with a “<” sign), equal to (with a “=” sign), and three logic gates (i.e., AND, OR, NOT). When setting the condition for the “If-Then” block, students might wrongly adopt inappropriate conditional expressions because they failed to understand the meaning of these conditional expressions. For example, some may use “>” to represent “<,” while some may get confused about “AND” and “OR.”

Problems in iterative logic. In Scratch, “Repeat,” “Forever,” and “Repeat-Until” blocks are available for achieving iterative logic. Iterative expressions are usually for looping, such as keeping a robot moving. However, some students might forget to use iterative expressions when they should; consequently, they might find that their robots moved for only a while and then stopped. Some may wrongly nest iterative logic in conditional logic; therefore, they may find that the robot did not move at all.

Problems in parallel logic. Parallel logic in Scratch programming is simply referred to as “multiple scripts running in parallel,” which is akin to multiple threads in programming.

Meanwhile, the concept of “multiple threads” is a more complicated and professional idea. For young-aged children, Scratch could also realize simply parallel logic, such as an “If-Then” block followed by another one or more “If-Then” block(s) in a looping block. Although the scripts would be read block-by-block, each “If-Then” block could be regarded as parallel in one loop. At the summer camp of this research, parallel logic was especially important for a complex task, such as creating a football robot, because a football robot had to detect and judge various conditions. However, in the classroom, many students raised questions about how to allow the robot to detect conditions A and B at the same time. Some may think that condition A followed by condition B means: B would be read only if A was true. Helping students comprehend parallel logic can advance their abilities to program for complex situations.

In previous studies, such as Leonard et al. (2016), computational thinking was regarded as a skill of algorithmically solving problems. However, with a tangible robot, young-aged children could solve programming problems not only algorithmically but also visually and tangibly. In this research, which was conducted in a robotics education environment, robot programming is different from other types of programming, such as software programming, which is based mainly on algorithm design or logical simulation, as stated previously. It naturally needs to transform the programming or algorithm problems into a tangible form by debugging with a tangible robot, especially for young-aged children in primary school.

Computational thinking process: A "three As" circle. As shown in Figure 7, a "three As" computational thinking process could be found in this research, involving: Abstraction, Automation, and Analysis. This process basically aligns with a frequently-cited model of computational thinking proposed by Lee et al. (2011), which was based on Wing's works (Wing, 2006, 2008, 2010). Lee et al. (2011) demonstrated what these three terms look like in the robotics domain. Next, an example of this research could demonstrate this process.

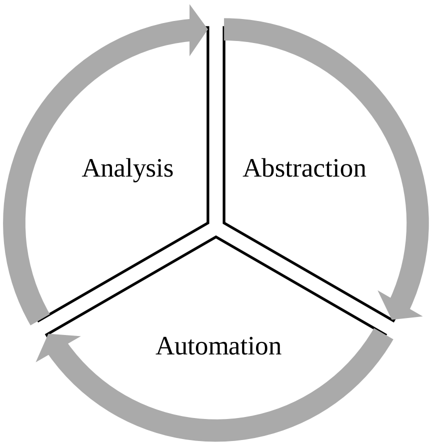
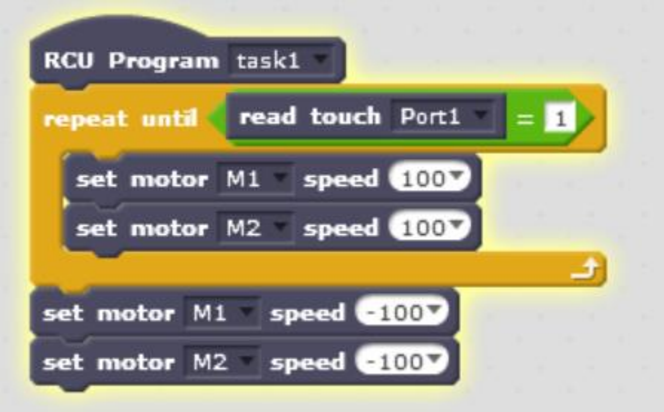


Figure 7. "Three As" Computational Thinking Process (Adapted from Lee et al., 2011).

Abstraction is programming for a robot by using various logics that are abstracted from the real world (Lee et al., 2011). In this stage, "students think about how to sense the world, and how those stimuli will be abstracted as numerical or true-false values inside the control program" (Lee et al., 2011, p. 34). In Lesson 4, the lesson task was to create a vehicle that could move and return when it touched a wall. Therefore, the conditions in this case could be described as: the vehicle touched the wall and did not touch the wall. How did students abstract these conditions? A student (B03) stated: “If the touch sensor receives a ‘1’ (TRUE) signal, then it means the vehicle touches the wall. If it receives a ‘0’ (FALSE) signal, then it means the vehicle does not touch the wall.” How did this student abstract the robot’s reaction in different conditions? He stated, “The motor speed should be a positive value [robot moves forward] until the signal [of the touch sensor] turns to ‘1’; then the speed should be a negative value [robot returns].” These abstracted statements were also presented in the students’ Scratch program by using conditional and iterative logics (as shown in Figure 8).



```
RCU Program task1
repeat until read touch Port1 = 1
  set motor M1 speed 100
  set motor M2 speed 100
set motor M1 speed -100
set motor M2 speed -100
```

Figure 8. Student B03’s Example Scratch Program Showing Conditional and Iterative Logics.

“Computing is the automation of our abstractions” (Wing, 2010, p. 2). In robotics, automation is the stage when the student downloads the program to the computing device (i.e., Intelligent Brick) and observes how the robot executes the program (Lee et al., 2011).

Although this stage is implemented mainly by the computing device, and students' direct participation seems inconspicuous, it is important for students to collect data by observing the running results (i.e., the robot's behaviours), which may influence the analysis stage. For example, in the case of making a touch-and-return robot, one student (B07) found that the robot did not stop or return when it touched the wall; he then attempted to run the program several times. When he collected enough information from the robot's mis-reactions, he went to the analyse stage.

In the analysis stage, by analysing information from the automation stage, students judge whether the robot behaved as expected (Lee et al., 2011). If the robot misbehaved, students may need to investigate the problem and return to the abstraction stage to analyse and refine the program. For the example mentioned in the automation stage, the student (B07) found that the robot did not behave as expected; therefore, he tried to find the reason. He analysed the robot's behaviours and said, "My robot moved forward to the wall firstly, so this part [of the program] is right, but it did not return [when it touched the wall], while I have already set the [motor] speed to negative. This means the robot did not know it touched the wall, so the problem might be related to [the setting for] the touch sensor." Then he went back to check his Scratch program and refined the conditional expression for the touch sensor.

The “three As” circle process of computational thinking resembles a real programmer’s programming and debugging process, although it is simpler and more matched to young-aged children’s learning level.

Relationship between CT skill and PS competencies. The results of the quantitative data presented before indicated a significant, strong positive correlation between students’ computational thinking skills and their problem-solving competencies (*rs*(30) = .64, *p* < .001). This relationship was not the focus of this research; however, through classroom observation, it was found that those who performed better in solving problems may also be better computational thinkers. This might be because computational thinking involves problem-solving skills when solving programming problems. For example, students in the automation stage of computational thinking may utilize the skills of generation of alternative solutions, decision making, and solution implementation. Also, the problem-solving process may involve students’ CT skills, such as using logical thinking when coming up with and evaluating solutions.

5.2.2 Development of Students’ Computational Thinking Skills: Qualitative Evidence

In quantitative data, the results of the Wilcoxon tests indicated that students’ CT skills (i.e., the sum score of six domains) significantly increased from the first six days (M_{first} = 7.326, Mdn_{first} = 7) to the last six days (M_{last} = 9.393, Mdn_{last} = 8.958), with *z* = -4.936, *T* =

528, and $p < .001$. As for the qualitative data, some evidence of students' computational thinking development was found.

Because the Computational Thinking Rubric adopted in this research is based on the computational thinking model proposed by the International Society for Technology in Education (ISTE) (2014), in qualitative data analysis, this ISTE model will be used to demonstrate evidence of students' computational thinking development. The ISTE (2014) outlined six components of computational thinking skills: formulating problems, abstraction, logical thinking, using an algorithm, analysing and implementing solutions, and generalizing and problem transfer. Next, corresponding evidence of students' computational thinking development will be presented.

Formulating problems. This component of CT skill refers to "formulating problems in a way that enables us to use a computer and other tools to help solve them" (ISTE, 2014, p. 1). With the emerging level of formulation problems, students may propose problems that are unrelated to the task goal or that are too general to be solved. For example, during the first week of the summer camp, when the teacher encouraged students to propose problems according to the lesson task, many students raised general problems, such as "How to make a successful robot" or "How to program." This situation was gradually improved with the progress of the summer camp. Students attempted to break down the general problems into smaller and more explicit problems, such as “How to order the robot to detect the wall/line” or “If I want to make the robot turn left, what parameter should I set for two motors?”

Abstraction. This component means “representing data through abstractions such as models and simulations” (ISTE, 2014, p. 1). With an emerging or moderate level of abstraction, students showed no evidence, or limited evidence, of interpreting problems in an abstractive way. For example, during the first week, because they usually formulated general problems, it was hard for them to abstract problems properly; therefore, they had to rely on the teacher’s sample program to complete the task. During the second and third weeks, they showed more evidence of abstraction; for example, they orally described problems by stimulating the robot’s reactions/behaviours, such as “the robot goes [forward], then it comes to a wall and then it touches the wall. How does it detect that there is a wall?” During the last week, some could abstract the robot’s reactions with Boolean logic (i.e., True or False, 1 or 0) and simulate the robot’s behaviours by following the flow of the program, so they could program independently.

Logical thinking. Logical thinking in a CT skill is “logically organizing and analysing data” (ISTE, 2014, p. 1). It may be hard for us to identify whether or not a child’s mental thinking is logical but, through their programming process and their interactions with the teacher’s cognitive apprenticeship teaching, we can gain information about their logical thinking. For example, during the first two weeks, some students' programs were illogically designed. This might have led to the robot's misbehaviour, such as going around in circles or misreacting to the input. When the teacher tried to scaffold students to articulate their thinking process, some could not logically describe it. However, during the last week, students performed better in logical thinking. For example, their programs followed a logical flow and they could describe their thinking process more logically, using terms like "because … so … otherwise…" and "if … then … else…." In an interview, a student (C04) who usually communicated with her parents about her daily learning said, "My parents told me that they found that my description of my robot and program is becoming more and more logical."

Using algorithm. Using algorithm in CT means "Automating solutions through algorithmic thinking (a series of ordered steps)" (ISTE, 2014, p. 1). In programming, algorithmic thinking can be observed as the extent to which the student adopted proper programming logics (i.e., conditional logic, iterative logic, parallel logic). With the learning progresses, students learned more about these logics and could become more experienced when using these logics to solve problems. For example, during the first week, the most frequently used algorithm was linear (i.e., blocks were executed block-by-block without logical statements from beginning to end and only once). During the second week, the students tended to use more logical blocks like "If-Then" and "Forever" but not skilfully.

During the third week, they could adopt these logical blocks proficiently, though the program was still at a novice level. During the last week, the students could properly use various logics and nest them together as experts.

Analysing and implementing solutions. This component of CT refers to "identifying, analysing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources" (ISTE, 2014, p. 1); this involves the debugging process (Leonard et al., 2016). Debugging is a process commonly used by real-world programmers to analyse the outcomes of implementing solutions and find faulty step(s) to refine the program. During the first week of the summer camp, students showed little evidence of debugging because they highly relied on the teacher's sample program when they failed to complete the task. During the second and third weeks, they could debug with the teacher's scaffolding and coaching. During the last week, some of them could debug independently even when accomplishing more complex tasks.

Generalizing and problem transfer. A better computational thinker can generalize and transfer the problem-solving process for a given problem to a wide variety of problems (Atmatzidou & Demetriadis, 2016; ISTE, 2014). For example, during the first week of the summer camp, some students may have often experienced a similar problem, such as "how to order the robot to move for a given time" or "how to order the robot to return." As learning progressed, some students gradually became able to transfer the problem-solving process to similar problems, but would be stuck by novel problems. During the last week, some students could utilize old knowledge or problem-solving experiences to solve new problems. For example, in the last lesson about making a football robot, a student (D03) used the solution in Lesson 5, which was to follow a black line, to solve the problem of following a light ball.

In summary, the observation of students' performance revealed the enhancement of CT in six aspects. Corresponding evidence was presented above. Also, in the interview, the teacher acknowledged the enhancement of students' CT skills. From the first interview (at the end of the second week):

Researcher: "How do you describe students' performance in the six domains of the rubric [of CT]?"

Teacher: "To be frank, when you gave me the rubric, I thought their performance might disappoint us. At least up to now, their improvements exist but are limited."

In the second interview, the teacher changed his mind:

"I witnessed their growth in CT. They can properly formulate problems. When solving problems, their abstracted and logical thinking could be seen. They also showed some basic idea of using an algorithm and debugging, although, as you know, it is hard for such young-aged children. And, they can transfer the problem-solving process to solve new problems."

Another important finding is that students' formulating problems, abstraction, logical thinking, and algorithm using were involved mainly in the *Abstraction* stage; Meanwhile, the *Automation* and *Analysis* stages mainly involved their abilities to analyse and implement solutions, as well as generalize and problem transfer.

It should be noted that although students' mean score in CT skills increased significantly from 7.32 to 9.39 (i.e., this is the total score of six components, so the average level of each component increased from 1.22 to 1.56), students' CT skills did not reach a "moderate" level (between 1 [emerging] to 2 [moderate]). Classroom observation revealed that while general improvement was acknowledged, an individual's variety should also be considered.

5.2.3 Effective Classroom Interactions in Students' Computational Thinking

Some evidence of students' CT skill development was found in this research, as presented previously. Nevertheless, behind that improvement, more details on students' learning process, which may explain the improved result, should be explored. According to the correlation analysis, students' computational thinking improvements were positively correlated to time spent on three interactions. In qualitative analysis, some effective classroom interactions in students' CT skill development were mainly focused upon.

Table 55

The CT Skill Measures of Four Focal Samples

	No.	Gender	Grade	First six days	Last six days	Difference
Kevin	A01	Male	2	7.667	11.167	3.500
Abby	B04	Female	2	7.000	9.917	2.917
	Average level			**7.326**	**9.393**	**2.067**
Terry	A07	Male	2	7.333	8.250	.917
Nancy	B09	Female	2	7.250	8.083	.833

Similar to analysing students' problem-solving competencies, when analysing interactions, another four focal samples were critically identified, including two whose measures of CT skill were obviously increased and two that were slightly increased. As shown in Table 15, the first-six-days scores of four individual focal cases were at a similar level (from 7.00 to 7.67). All were grade-2 students. Next, through the process of tracking their interactions with the teacher, peers, and robot, from the first class to the end, the typical kinds of interactions that might explain their different development levels of CT skill will be characterized.

Effective S-T interactions in students' CT skill development. Quantitative analysis identified a positive correlation between students' time spent on S-T interactions and a CT skill measure increase ($rs(30) = .60$, $p < .001$). As mentioned before, the teacher adopted cognitive apprenticeship teaching methods at the summer camp. With the teacher's modelling, coaching, scaffolding, articulation, reflection, and exploration methods, students' problem-solving competencies were enhanced. As for CT skill development, the teacher's cognitive apprenticeship teaching was also effective. Moreover, students' active engagement interactions with the teacher were found to be effective for their CT skill development.

Cognitive apprenticeship teaching methods. This section will demonstrate how the teacher performed cognitive apprenticeship teaching methods, including modelling, coaching, scaffolding, articulation, reflection, and exploration, and how students benefited from this teaching process.

modelling. The teacher engaged in modelling by orally formulating problems and drawing the flow chart of the programming process. For example, in Lesson 3 (i.e., making a robot move forward for a given time and then return), the teacher encouraged students to formulate possible problems based on the lesson task. Because students' ability to formulate a problem was at a lower level, their formulated problems were sometimes too general. Therefore, the teacher shared his formulated problems with students: "The purpose of making this robot is to let it move forward for, let's say, 10 seconds and then return to the beginning point … So, the programming problems here can be: First, how to order it to move forward? Second, how to order it to move for 10 seconds? Third, how to order it to return? Fourth, how to order it to return and move for another 10 seconds?" After this, the teacher drew a flow chart of the programming process, addressing problems above. Following the teacher's modelling, students were gradually guided to formulate problems and abstract like experts. In the interviews, many students thought that the teacher's flowcharts were useful for them. For example:

Student B06: "When I started to program, I often referred to what the teacher draws on the blackboard [the flowcharts]."

Student B05: "Yes, he clearly showed us how to program, so we can imitate his steps [of programming]."

However, it should be noted that some students totally relied on the teacher's flowcharts, so that they cannot program independently. For example:

Student C06: "What the teacher draws on the blackboard are useful…but if he didn't show us those stuffs [the flowcharts], we wouldn't be able to program."

Student A07: "…I don't think we can program without the teacher's drawings [the flowcharts]."

Also, the teacher expressed his concerns about this situation:

"I drew the flowchart in order to make students better at programming, but actually, I found that they seem to be tied to those flowcharts. In the first week, I drew the flowcharts in detail, so they can basically complete the programming tasks; while since the second week, I

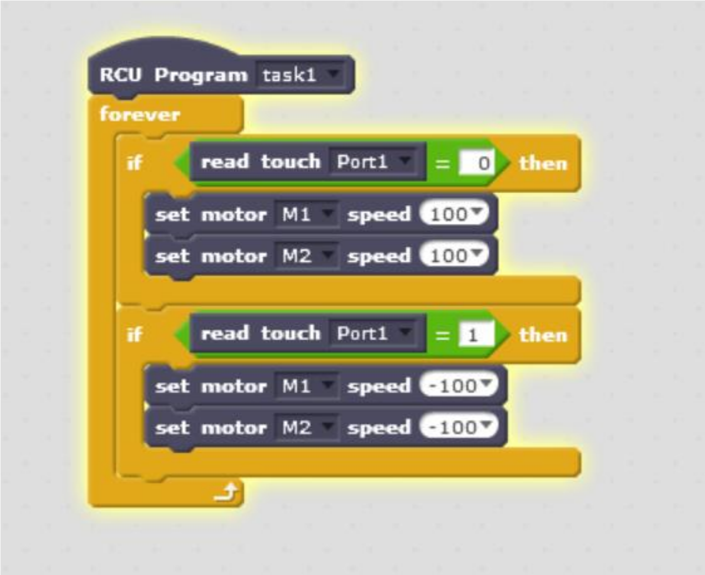
deliberately hid some information in the flowcharts, and consequently, some students got lost.''

This finding reminds us about two issues. Firstly, the purpose of modelling is allowing students to generate their own conceptual model, so it is important for the educator to carefully diagnose whether students have built their own conceptual model. Secondly, this finding also reminds us about the importance of the articulation activities. By encouraging students to articulate their thinking process, the teacher can know whether students have ability to formulate and abstract programming problems independently.

coaching. When students were programming, the teacher observed and coached them by providing suggestions, reminders, and support. Because participants in this research were all new to programming and it was hard for such young-aged children to manipulate the programming interface (i.e., Scratch), the teacher's coaching was important for students' abstraction, logical thinking, and algorithm use. For example, in Lesson 4, students learned about the mechanism of the touch sensor, while in programming, some may have failed to abstract the mechanism to programming. Therefore, the teacher often reminded them, "if the button of the touch sensor is pressed, then it will receive a '1' value; otherwise, it will receive a '0' value." When some children's robots were out of control, the teacher coached them to logically think about where the problem was in their programs. In addition, based on abstraction and logical thinking, the teacher coached students to select suitable algorithm(s) for logically stating abstracted problems in programming.

scaffolding. The teacher's scaffolding was mainly by providing sample program(s) and using the internal-test model of Intelligent Brick. The sample program was not a program that had already been successfully completed, as providing a given successful sample for students may have resulted in simple copying rather than critical analysis and creative programming. Therefore, sample programs were related to the lesson task but not yet completed. For example, there were several samples related to controlling motors, while some other samples related to controlling the touch sensor. To complete the task of Lesson 4, students had to critically analyse which samples were suitable for Lesson 4 and how to integrate and modify them to achieve the purpose. This kind of scaffolding involved students' abilities for logical thinking, algorithm using, and debugging. Moreover, scaffolding with Intelligent Brick's internal-test model may help promote students' CT skill development. By using the internal-test model, the teacher can demonstrate how motors and sensors react to the real world. For example, in Lesson 11, which involved making a robot follow the people's tracking, an ultrasonic sensor was used to measure the distance between a robot and an object. The teacher used the internal-test model to demonstrate how a robot can detect a person in front of it: "As you can see, when the robot gets closer to an object, the measure of ultrasonic was smaller; when the object moved far away from the robot, the measure became larger." By doing this, students could visually see how the robot detected the real-world situation, which allowed them to logically abstract the robot's reaction into programming.

articulation. Because programming problems were usually abstracted, the teacher's articulation may have made students' tacit thinking process more concrete and explicit. When students were debugging their programs, the teacher encouraged them to simulate the robot's routine orally or physically, so the students could easily find the faulty part of the program. For example, in Lesson 4, a student (C03) wrote a program as shown in Figure 9. In his mind, the program seemed logical. However, when this program, was executed, the robot misbehaved. Therefore, the teacher encouraged the student to simulate the robot's behaviour with this program. The student pointed at the computer screen and said, "If the touch sensor received a '0' value, then the robot will move forward. If the touch sensor received a '1' value, then the robot will return." The teacher reminded the student: "Your program has a 'forever' block. Then what will the robot do?" The student said, "The robot will repeatedly judge whether the touch sensor receives a '1' or '0' value." The teacher encouraged the student to handle the robot to simulate the robot's actual routine. The student moved the robot using his hands and said, "It moves forward to the wall … It touches the wall, so it returns … Oh, I see what the problem is … When the robot returns, the button of the touch sensor was released, so it received a '0' value. Then it will move forward again." As can be seen from this case, if students' thinking was not articulated, it would be hard for the teacher to know how to help them. Then the students' abilities for abstraction, logical thinking, algorithm using, and debugging (analysing and implementing solutions) might not be well-developed.



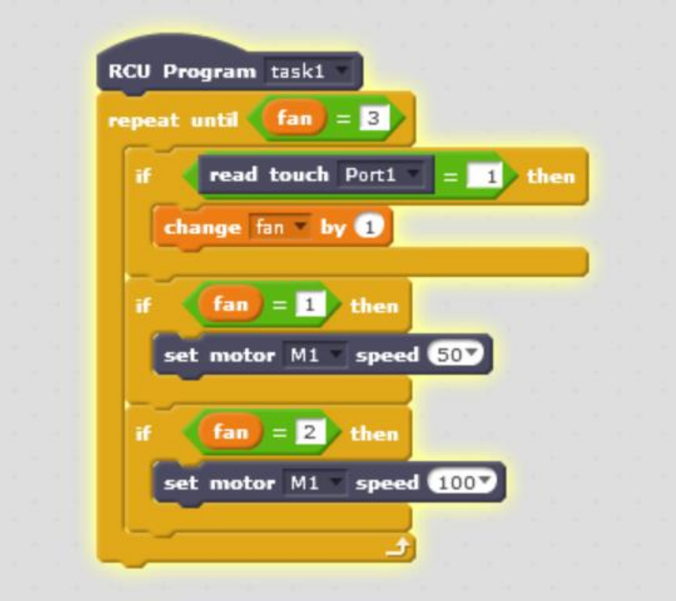
```
RCU Program task1
forever
    if read touch Port1 = 0 then
        set motor M1 speed 100
        set motor M2 speed 100
    if read touch Port1 = 1 then
        set motor M1 speed -100
        set motor M2 speed -100
```

Figure 9. A Student (C03)'s Program for Lesson 4.

reflection. Reflection allowed students to compare their computational thinking process to an expert's. For example, by reflecting on formulated problems, students learned to break down a general problem into smaller ones; by reflecting on the abstraction process, students learned to interpret problems in a more abstracted way; by reflecting on the programming process, students learned to think more logically and adopt a more proper algorithm; and by reflecting on the debugging process, students learned to implement and analyse solutions with computational thinking.

exploration. Exploration allowed students to generalize and transfer the computational thinking process from a given task to a more challenging and creative task. For example, in Lesson 7, students explored more functions of a fan, such as a fan with different turning speeds controlled by the touch sensor. Figure 10 showed a student's (A06) program, which was for making the fan turn at different speeds. The techniques used in this program were what the students had already learned, but they creatively used these techniques to explore solutions to the new challenging task.

To conclude, the teacher's cognitive apprenticeship teaching, including modelling, coaching, scaffolding, articulation, reflection, and exploration, may benefit students' CT skill development. Nevertheless, students' engagement in the teacher's teaching process might also influence their CT skill development.



```
RCU Program task1
repeat until fan = 3
    if read touch Port1 = 1 then
        change fan by 1
    if fan = 1 then
        set motor M1 speed 50
    if fan = 2 then
        set motor M1 speed 100
```

Figure 10. A Student (A06)'s Program for Controlling the Fan to Turn in Different Speeds.

Students' engagement in cognitive apprenticeship teaching. Students' CT skill development varied, though they were all involved in the same teacher's cognitive apprenticeship teaching. To further explain various student results, students' learning interactions should be concerned. In this research, by characterizing those whose CT skill was obviously enhanced and those whose CT skill was slightly improved, the findings showed that whether students were positively and actively engaged in S-T interaction was significant to their computational thinking skill. Through videotaped data, the performances of four focal samples were tracked, as introduced previously. Their cases will be used to demonstrate this finding.

active engagement. Kevin is a grade-2 boy whose CT skill measure increased by 3.50 (M = 2.07) from the first six days to the last six days. Namely, he gradually became a better computational thinker. The process of tracking his performances during the summer camp identified his active engagement in S-T interactions. For example, when the teacher was modelling by drawing the flow chart on the blackboard, Kevin dedicatedly followed the teacher and sometimes could orally describe his own flow chart before the teacher's drawing, although he was not always right. During the first week, his CT skill was usually at an emerging level, so he tended to formulate general problems and failed to abstract problems. However, Kevin proactively shared his thinking process with the teacher, so the teacher could understand his weaknesses and accurately coach or scaffold him.

Moreover, Kevin tended to pose some constructive questions to the teacher. During the first two weeks, many students asked the teacher for help with superficial questions. However, the questions Kevin raised seemed more constructive. For example, in Lesson 5, when making a line-following robot, many students asked the teacher about whether the lighting sensor should be settled on the right side or the left side of the line. For this question, both right and left answers were right; this was not the point of making a line-following robot. The point was, once one decided on which side to place the lighting sensor, how would this arrangement cooperate in the programming? In other words, deciding which side is "right" depending on one's preference; therefore, when responding to this question, the teacher's answer might be limited to "it depends on your preference." However, Kevin's question was more constructive: “I decided to put the lighting sensor on the right side of the line, so the robot would turn left first to detect the line. If the lighting sensor ‘sees’ the black line, then the robot should turn right, but what [conditional expression] should I set for the ‘If-Then’ block? Do I need to use the ‘If-Then-Else’ block? What is the difference [between the two blocks]?” Consequently, the teacher coached him about how to logically state a conditional expression and how to select proper algorithm logics. From this case, we can see that, when a student asks questions that are too superficial, the teacher’s response might also be simple; however, when a student asks constructive questions, the teacher can respond more to the student and provide more support for the student’s CT development. This finding reminds us that, although the time spent on S-T interactions was statistically correlated with students’ learning outcomes, it is just one side of the story. How students interact with the teacher should also be considered. In this research, students’ active engagement in S-T interaction means not only spending much time interacting with the teacher but also effectively engaging in S-T interactions.

inactive engagement. Nancy is a grade-2 girl whose CT skill measure only slightly increased by .83 (*M* = 2.07) from the first six days to the last six days of the camp. By tracking her performance during the summer camp, her inactive engagement in S-T interactions could be found. For example, she replied to the teacher’s step-by-step coaching and scaffolding. When the teacher was coaching other students and had not yet come to her, Nancy would not start to program. When the teacher came and supported her, she did what the teacher said, but without critical thinking and questioning. Many times, she even asked the teacher to directly create a successful program for her. When the teacher tried to scaffold her by showing sample programs, Nancy simply copied those programs that she thought to be useful and gathered them illogically. Therefore, when the lesson task was simple, her works might have been successful, though when the lesson task was more difficult, she could not program independently. In this case, Nancy's interactions with the teacher failed because the teacher's supportive behaviours were unidirectional, without students' active engagement. S-T interactions should be a bidirectional process involving both the teacher's teaching and the students' active engagement in learning. Otherwise, as can be seen in Nancy's case, the student would rely on the teacher's step-by-step guiding and be unable to think independently.

Reflecting on the cases of Kevin and Nancy, we can see that CT skill development requires both the teacher's cognitive apprenticeship teaching and students' active engagement in S-T interactions. Without students' active engagement, the teacher's unidirectional teaching would be ineffective. Students' active engagement may allow the teacher to provide targeted support more accurately, while inactive engagement may hinder the teacher's support or lead to over-reliance on the teacher's guidance.

Effective S-S interactions in students' CT skill development. As mentioned previously, although no group/pair-formed activity was involved in the summer camp, S-S interactions still existed. From the quantitative data, a strong positive correlation was identified (*rs*(30) = .64, *p* < .001). However, the kinds of S-S interactions that may effectively lead to students' CT skill enhancement should be further characterized. In the original conceptual framework of this research, cooperative interactions (i.e., Discussion and observational learning) and off-task interactions were included. However, through the analysis of qualitative data, no sufficient evidence supported the idea that off-task interactions were effective. In the process of tracking four focal samples' performances from the first class to the end of the summer camp, similar to the finding of problem-solving competencies, cooperative interactions were identified as the main kinds of effective S-S interactions; those who cooperative well with peers may have performed better in computational thinking. Two cooperative interactions were characterized as effective in the development of CT skills: discussion and observational learning.

Discussion. When students were programming on their own computers, the teacher coached and scaffolded them one-by-one; therefore, most of the time, when the teacher focused on one student, other students discussed programming with their peers. The focal sample Alice, for example, usually exchanged her ideas with others. For example, in Lesson 5 (line-following robot), the following exchange occurred:

Alice: "Do you have any ideas about programming?"

Peer: "Not yet, but at least we should know how the robot sees [detects] the black line."

Alice: "Yes, as the teacher said, the lighting sensor may receive different values when it sees different colours. I think that is how the robot distinguishes between the black line and the ground."

Peer: "Then how can we make it move forward while it is searching [for the black line]?"

Alice: "Setting positive values for the [speed of] motors?"

Peer: "For both motors?"

Alice: "I'm not sure about that…"

Peer: "Do you remember what the teacher's [sample] robot looks like? His robot repeatedly vacillated to the left and right."

Alice: "Oh, yes. Then the values setting for two motors should be different. If it turns to the right, the left motor should be 100 [speed] and the right motor should be 0. If it turns to the left, the right motor should be 100 and the left motor should be 0."

Peer: "Yes. And if the lighting sensor detected that there is no line, then the robot turns to the left; otherwise, it turns to the right."

Alice: "Then the 'If-Then' block should be used…"

Peer: "Two 'If-Then' [blocks] and one 'Forever' [block], I think. Because one 'If-Then' is for turning left and another is for turning right. 'Forever' made it repeatedly detect [the black line]."

Alice: "I think we can use one 'If-then-else' instead of two 'If-then.' It's simpler."

Peer: "You are right. It's better."

In this case, we can see that, Alice and her peer exchanged ideas and sometimes reminded each other about information provided by the teacher. By doing this, they reviewed the teacher's modelling and logically gathered information from the teacher to use in their own designing. Through the discussion process, by thinking aloud to each other, they abstracted the problem more concretely. Also, through discussion, they could recognize the parts for which they had to seek help from the teacher, which meant that the support that the teacher provided to them might be more accurate.

Beyond this case, S-S discussion also showed other advantages that may lead to CT skill enhancement. For example, in algorithm using, a peer's reflective discussion may provide useful information about which algorithm logic is more suitable to the lesson task. In analysing and implementing solutions, discussion allowed students to share their debugging process with others. In generalizing and transferring the problem-solving process, discussion may remind students about the possibility of using a known problem-solving process to solve new problems.

In the case of Terry, whose CT skill measure slightly increased by .92 (*M* = 2.07), he rarely participated in S-S discussion. Consequently, he was usually unable to solve some simple programming problems, as he did not share and discuss his faulty ideas with others. The teacher's coaching and scaffolding for him usually started with the teacher helping Terry to address the basic concepts of programming. Therefore, his learning progress was slower than that of others and there was no room for him to achieve higher-level computational thinking. While sometimes he could complete the lesson task with the teacher's support, Terry's reflection on his work was usually sparse, as he could not build his own understanding of programming without peer sharing and discussion.

Observational learning. Students' observational learning was also significant to their CT skill development, as they learned from others' failures and successes. For example, Kevin was a proactive observer in the programming process. When one student initially finished programming and was starting to debug, Kevin carefully observed the student's debugging process and sometimes talked with the student about programming or debugging techniques. Moreover, by critically observing others' failures and successes, he learned how to modify his own program. In the interview, Kevin described his observational learning process:

> *Researcher: "Do you have difficulties with programming?"*
>
> *Kevin: "Of course. Everyone does."*

Researcher: "What would you do to tackle these problems?"

Kevin: "I saw how other classmates did things. If their programs failed, then I knew their ideas might be wrong, so I would avoid the mistakes they made. If someone's program was successful, then I would learn from him."

Research: "What do you mean by learn from him'?"

Kevin: "I modified my own program according to his ideas."

Research: "So, you did not simply copy his program?"

Kevin: "No. I just modified my own program."

From Kevin's case, we can see that observational learning allowed the student to compare his own work to that of others; therefore, students learned how to succeed in programming and how to program and debug like "experts." However, in the case of Nancy, she focused only on her own work and rarely observed peers' work. Therefore, she lost the opportunity to learn from others and to reflect on her work.

What should be noted is that observing did not mean copying. Observational learning requires students' critical thinking to distinguish which parts of others' work could be adopted and which parts could not. Meanwhile, copying is to simply accept others' ideas and directly use other's programs. As compared to Kevin's case, Terry's "observation" behaviours were more like "copying." For example, as found in the videotapes, Terry usually wrote his program by watching others' completed programs while the teacher was not looking at him. However, sometimes, the program he copied was not yet debugged, so his robot with a faulty program did not successfully complete the lesson task. As a result, when the teacher came to coach and scaffold Terry, he could not explain his programming process. These led the teacher's teaching to be inefficient.

In addition to discussion and observational learning, students' off-task interactions were analysed. However, it was found that off-task interactions were ineffective in developing students' CT skills. The reason is the same as that provided for PS competencies: Students might get lost in off-task interactions. Because the development of CT requires students to focus on solving programming problems, off-task interaction might distract students; consequently, they get lost in off-task activities and lose the chance to develop thinking abilities.

Effective S-R interactions in students' CT skill development. S-R interactions played an important role in students' abstraction—the foundation and essence of computational thinking (Wing, 2006, 2008)—because the robot is a tangible and visible agent between the real world and the abstracted computing world (Bers et al., 2014). Interacting with a robot helps students abstract the real-world situation to the programming process, which could benefit students' CT skill development. In quantitative analysis, a significant positive correlation was found between students' time spent on S-R interactions and their CT skill development (*rs*(30) = .51, *p* = .003). In the original conceptual framework of this research, building, programming, observing, and participatory investigation were included in S-R interactions, while in qualitative analysis, the critical selection of inputs, observational investigation, and participatory investigation were found to be effective in promoting students' CT skill development.

Critical selection of inputs. The selection of inputs is a process of "talking" to the robot by "manipulation of variables and computational instructions" (Bers, 2010, p. 3). Computational abstraction relies on the selection of inputs (Bers, 2010) and the selection of inputs is for abstracting the expected movements of the robot to programming. For example, in Lesson 5 (line-following robot), to order the robot to move by following the black line, students had to select suitable algorithm logics and decide which logical instructions to input to the robot.

It was found that those who critically selected inputs may achieve a higher level of CT skill. For example, when Kevin was programming, he firstly browsed given sample programs and then simulated the possible consequences of different programs. After that, he selected and gathered suitable scripts for his own program. Also, he selected suitable conditional expressions and decided on the proper parameters for setting conditions by using the internal-test model of Intelligent Brick. As can be seen from this case, the process of selecting inputs involved students' abilities for abstraction (such as simulation), logical thinking, and algorithm using. However, in the case of Terry, he tended to program without carefully and critically selecting inputs. Often, he started to program by using his personal preference and habits. For example, during the first week, he used a linear algorithm that was suitable for the lesson tasks; however, during the second week, he still adopted a linear algorithm that was not suitable for the new lesson tasks. When the teacher asked why he used the linear algorithm again, he said: “Because I got used to programming like this.” This means that he did not critically select inputs when the situations changed. Therefore, his abilities for abstraction, logical thinking, and algorithm using could not be stimulated.

Observational investigation. Observational investigation is the process of “listening” to the robot’s behaviours. Computational abstraction also relies on the observation of outputs (outcome data) (Bers, 2010). In pure programming, the outputs are visible on the computer interface but are not tangible, while in robotics programming, the outputs can be observed from the robot’s behaviours, which are quite tangible; therefore, it is easier for students to observe and analyse. Observational investigation may facilitate students’ ability for abstraction and solution implementation and analysis. For example, Kevin often debugged a program by observing the robot’s behaviours. If the robot misbehaved, he carefully observed the situation to find the problem.

What should be noted is that observational investigation is more than just watching what happens. For example, Terry was also watching his robot’s performance, but he usually ignored the robot’s misbehaviours and did not seriously regard unexpected outcomes. Therefore, sometimes he did not know how to revise his program until the teacher or his peers pointed out the faulty part; other times, he got lost in laughing at the robot’s strange behaviours.

Participatory investigation. As mentioned previously, *participatory investigation* is a kind of direct bodily S-R interaction involving programming and debugging in which “the child’s role shifts from designer and observer to that of participant” (Levy & Mioduser, 2010, p. 28). From our observations, almost every student was willing to bodily play with the robot, such as blocking, changing, or follwing the robot’s path. However, not all of them could gather and investigate information from playful bodily S-R interactions. For example, Nancy played with the robot frequently; however, she simply enjoyed the playing process, even when the lesson task had not yet been completed. In fact, bodily participation in the robot’s behaviours may result in the gain of much useful information for programming and debugging. For example, in Lesson 4, when Alice was programming and debugging, she held her robot in her hands and pressed the button of the touch sensor to see whether her program worked. Therefore, she did not need to automate the robot on the ground many times. In Lesson 5 (line-following robot), she stepped following the robot’s routine; when the robot got lost, she used her hands to correct the robot’s routine and see in which part the robot would get lost. In Lesson 12 (football robot), Alice moved the lighting ball by hand to simulate the ball’s movements so that, in programming, she could consider more possible conditions when searching for the football (lighting ball). By participatory investigation, as mentioned above, students debugged their programs so that they became more logical in a concrete way; thus, their CT skills were developed.

In conclusion, S-R interactions, including the selection of inputs, observational investigation, and participatory investigation, were effective in promoting students’ CT skill development. These kinds of S-R interactions allowed students to think by doing, by bridging abstracted program designing and debugging to concrete robot behaviours. The selection of inputs is a process of “talking” to the robot, while observational investigation is about “listening” to the robot. This bidirectional process showed what mutual interactions between student and robot look like and how students can benefit from this process. In addition, participatory investigation showed the potential of bodily interacting with the robot. Students gained useful information from bodily S-R interactions, which may effectively promote students’ programming and debugging abilities and further enhance students’ CT skills.

5.2.4 Summary

The findings showed three features of computational thinking. First, CT skill focused on solving programming problems, including conditional logic, iterative logic, and parallel logic. Second, children’s CT process generally followed a “three As” circled process, including abstraction, automation, and analysis. Third, CT skill is positively related to problem-solving competencies. Also, evidence showed that students' computational thinking skill developed during the summer camp in six aspects: formulating problems, abstraction, logical thinking, using an algorithm, analysing and implementing solutions, and generalizing and problem transfer.

Moreover, three kinds of interactions (i.e., S-T, S-S, and S-R interactions) played important roles in the development of students' CT skills. How these interactions influenced students' CT skills was presented. In S-T interactions, the teacher's cognitive apprenticeship teaching methods and the students' active engagement contributed to the student's CT skill development. With cognitive apprenticeship teaching methods, students could think aloud, which benefits the students' CT skills if the students actively engaged in the teacher's teaching process. Among S-S interactions, discussion and observational learning may help develop students' CT skills because: a) Through the discussion process, by thinking aloud to each other, they abstracted problems more concretely. Also, by discussing, they could recognize for which parts they needed help from the teacher; thus, the teacher's support could be more accurate. Moreover, through peer sharing and discussion, a student could build his own understanding of programming. b) Observational learning allowed students to reflect on their own works; therefore, they might learn how to succeed in programming and how to program and debug like "experts." As for S-R interactions, the selection of inputs, observational investigation, and participatory investigation were characterized as three kinds of S-R interactions that may influence students' CT skill development. These kinds of S-R interactions allowed students to think by doing, by bridging abstracted program designing and debugging to concrete robot behaviours.

In addition to the findings above, we should note that these interactions often worked together to contribute to students' CT skills. For example, for purposes of formulating problems and abstraction, actively engaging in the teacher's cognitive apprenticeship teaching, together with discussion with peers, might help students break down the general problems and abstract real-world situation to the programming process. For logical thinking and algorithm using, interacting with the teacher may contribute to students' initial understanding of programming, while through S-S interactions and S-R interactions, students could build their own comprehension.

Chapter 6: Discussion and Conclusions

In this chapter, the quantitative results and qualitative findings presented previously will be discussed to answer the research questions. Additionally, the revised conceptual framework will be presented. Also, the contributions and implications of the results/findings will be demonstrated. Moreover, the limitations of this research will be stated. Lastly,

6.1 RQ1: What Are the Features of Students' Problem-Solving and Computational Thinking in a Robotics Learning Environment?

To answer this research question, the qualitative data collected from semi-structured interviews and classroom observation were analysed. Several features of students' PS and CT were found.

6.1.1 The Features of Students' Problem-Solving

The findings showed that there were three features of students' problem-solving in a robotics learning environment (i.e., a robotics summer camp in this research). Firstly, in robotics learning, two kinds of problems were involved: building problems and programming problems. Secondly, students' problem-solving process was a five-stage circle. Thirdly, students preferred to intuitively come up with solutions.

Two kinds of problems in a robotics learning environment. Many previous studies stated that robotics learning involves problem-solving (Barak & Zadok, 2009; Castledine & Chalmers, 2011; Chang et al., 2010; Liu et al., 2013; Nelson, 2012; Zainal, Abdullah, & Prabuwono, 2012), while few tried to characterize problems in robotics learning. Some studies sought to distinguish lesson tasks from problems, while others did not distinguish building problems from programming problems, such as Barak and Zadok (2009). Some others discussed problem-solving regarding lesson tasks as problems, such as Castledine and Chalmers (2011). However, Castledine and Chalmers (2011) divided the inquiring tasks required to solve robotics problems into "robot design and construction, and software programming" (p. 23), which indicated emerging ideas about separating lesson tasks into building and programming processes. Therefore, it is significant to characterize different kinds of problems in robotics learning, as building and programming are two facets of creating a robot, and these two kinds of problems involve different thinking processes. This research found that, unlike with solving daily personal problems or academic problems (i.e., mathematics or calculating problems), there were two typical kinds of problems in robotics learning: problems about building a robot (*building problems*) and problems about programming for a robot (*programming problems*).

The findings showed that building problems are tangible and touchable, and that students can tackle them visually, while programming problems, to some degree, are nonobjective and students must solve them by logical thinking or transforming them into tangible forms. Therefore, when one is coaching or scaffolding students to solve different kinds of problems, the focus could be different. For example, in solving building problems, the focus of teaching is on how to properly assemble elements; meanwhile, for solving programming problems, the focus is on how to order commands logically. Also, both student and teacher must recognize and think about the kind of problems they are encountering, as the same situation might be the result of different kinds of problems. For example, if the robot does not react to touching the wall, it could be a building problem (i.e., the touch sensor was placed too deeply inside the robot) or a programming problem (i.e., the wrong conditional expression was set to the touch sensor).

A five-stage circle of students' problem-solving process. There are some typical problem-solving stages mentioned in previous studies. For example, as proposed by D'Zurilla and Goldfried (1971), a five-stage model includes general orientation, problem definition and formulation, generation of alternatives, decision making, and verification. To solve mathematics problems, Polya (2004) proposed a four-step model, including: understand the problem, devise a plan, carry out the plan, and look back. For children's problem-solving, for example, as found by Ömeroğlu and colleagues (2009), there are six stages of problem-solving: "being aware of the problem, defining the problem and understanding the reason, evaluating the reasons and guessing, collecting data to prepare for the solution, finding alternative ways of solution, and approving the ways of solution and testing them" (p. 1979).

For students' problem-solving in the context of learning robotics, for example, as suggested by Varnado (2005), a four-stage model includes: clarification, developing a design, modelling/prototyping, and evaluating the design solution.

In the current study, a five-stage circled model was found, which basically aligns with previous models of problem-solving as mentioned above. The first stage is to identify the problem; students define and formulate the problem they are going to solve. Based on the problem-solving goal, students come up with one or more solution(s) and decide which solution to carry out. After carefully carrying out the selected solution, students evaluate the outcome; if the solution failed, they return to a review and repeat the previous problem-solving stages until the problem is successfully solved.

The *circled* process is the key feature of students' problem-solving because it represents a repeated process of trial-and-error. In Barak and Zadok (2009), cycles of the trial-and-error process emerged early in students' period of learning robotics; when the students gained more experience, they tended to use several *heuristic search* methods (Hayes, 1978) to generate inventive solutions. However, because the participants in this research were young (i.e., the participants in Barak's study were junior high school students) and because the duration of the summer camp was shorter (i.e., the duration of Barak's study was three years), the heuristic search of students was sparse but emerged in a limited number of students. The potential of students' trial-and-error process in terms of developing heuristic search methods should be further probed in future studies in a long-term robotics learning environment. Additionally, the stage of finding an alternative solution aligns with the "redesign" stage of the engineering design process, which is also the key stage of engineering design (Eguchi, 2012).

Another finding should be noted: Students did not always find alternative solutions independently. Sometimes, they observed peers' solutions and were inspired by them or directly sought help from peers and the teacher, particularly in the early classes. This was because they were too young to independently find alternative solutions, especially when they were new to robotics. Nonetheless, in the later classes, their independence in generating solutions increased when they were equipped with more knowledge about robotics and had more experience with problem-solving.

Intuitively generate solutions. In this research, the findings showed that children's solution-generation process was usually intuitive, which aligns with the conclusion of Barak and Zadok (2009). Through intuitive thinking, some inventive solutions were generated. Nevertheless, the current study also found that students' intuitive thinking was not purely without foundation. Most of the time, the solutions intuitively generated by students were based on their own daily life experiences. This finding reminds us about Piaget's constructivism theories, which acknowledged that students' prior experiences may contribute to new knowledge construction, and Polanyi's tacit knowledge theories, i.e., "we can know more than we can tell" (Polanyi, 2009, p. 4). According to Piaget's constructivism theories, the teacher should encourage students to apply their life experiences to their problem-solving in robotics learning, such as by encouraging a student to generate solutions through a review of their own life experiences. According to Polanyi's tacit knowledge theories, the teacher should encourage students to think aloud when solving problems, such as by adopting cognitive apprenticeship teaching methods.

6.1.2 The Features of Students' Computational Thinking

The findings showed that there were three features of computational thinking: a) it focuses on programming problems, b) it follows a "three As" circled process, and c) computational thinking skill is related to problem-solving competencies.

Focusing on solving programming problems. Problem-solving competencies were discussed when students solved both building and programming problems at the robotics summer camp, while computational thinking skills were for solving programming problems. Previous studies concerning CT skill in robotics education (Atmatzidou & Demetriadis, 2016; Bers, 2010; Bers et al., 2014; Chen et al., 2017; A. Ioannou & Makridou, 2018; Leonard et al., 2016) evaluated students' CT skill through their programming-related performances. This was because the concept of computational thinking was based mainly on computing and programming environments (Wing, 2006, 2008; Yadav, Zhou, Mayfield, Hambrusch, & Korb, 2011). That is why this research emphasizes distinguishing between building problems and programming problems. When one is solving different kinds of problems, different abilities or skills are involved.

Programming problems in a robotics learning setting also have their own features. The first one is that robotics programming problems could be solved tangibly. In previous studies, such as Leonard et al. (2016), computational thinking was regarded as the skill of algorithmically solving problems. However, with a tangible robot, young-aged children could solve programming problems not only algorithmically but also visually and tangibly. In this research, which was conducted in a robotics education environment, robot programming is different from other types of programming, such as software programming, which is based mainly on algorithm design or logical simulation, as stated previously. It naturally must transform the programming or algorithm problems into a tangible form by debugging with a tangible robot, especially for young-aged children in primary school. This finding reminds us to teach programming by transferring abstract algorithms to tangible forms with robot kits. Also, tangible robots could help students connect reality to programming—namely, the abstraction process, which is the essence of CT skills (Wing, 2006, 2008).

The second feature of programming problems in robotics learning is that they stem from basic programming logics. Specifically, at the summer camp of this research, which was for young-aged children, programming problems usually stemmed from three typical programming logics: *conditional logic* (such as the If-Then block in Scratch), *iterative logic* (such as the Repeat/Forever blocks in Scratch), and *parallel logic* (such as multiple If-Then).

These programming logics were also deemed to be component elements of computational thinking in early literature (such as in Maloney et al. (2008)). Helping students understand the differences among these logics and adopt these logics properly is the main task of teaching CT skills. Understanding these logics allowed students to formulate suitable problems related to accomplishing the task, logically abstract problems in their programs, use the proper algorithm, debug accurately, and generalize the problem-solving process. This finding reminds us about the importance of demonstrating programming logics (including, but not limited to, the three logics mentioned above) when facilitating students' CT skills.

Computational thinking process: A "three As" circle. A "three As" computational thinking process was found in this research, including: Abstraction, Automation, and Analysis. This process basically aligns with a frequently-cited model of computational thinking proposed by Lee et al. (2011), which was based on Wing's works (Wing, 2006, 2008, 2010). Lee et al. (2011) demonstrated what these three terms look like in the robotics domain, and the findings of this research corroborate his model. Abstraction is programming for a robot by using various logics abstracting from the real world (Lee et al., 2011). In this stage, "students think about how to sense the world, and how those stimuli will be abstracted as numerical or true-false values inside the control program" (Lee et al., 2011, p. 34). "Computing is the automation of our abstractions" (Wing, 2010, p. 2). In robotics, automation is the stage when the student downloads the program to the computing device (i.e., Intelligent Brick) and observes how the robot executes the program (Lee et al., 2011). By analysing information from the automation stage, in the analysis stage, students judge whether the robot behaves as expected (Lee et al., 2011). If the robot misbehaves, students may need to investigate the problem and return to the abstraction stage to analyse and refine the program.

Although the problem-solving process (i.e., the five-stage circle) could be used for solving programming problems, this research also introduces how students used the computational thinking process to solve programming problems. Two processes are not conflicted, because the computational thinking process is just another version of the problem-solving process which focuses more on students' abstraction process. The "three As" model starts with the abstraction stage, which is similar to the initial stages of the problem-solving process (i.e., identify the problem and come up with a solution), though the abstraction stage of CT emphasizes formulating and representing problems in a more abstract way. In the second stage of the CT process, automation could be deemed as the "carry out the solution" stage of the problem-solving process. The analysis stage of the CT process also aligns with the last stages of problem-solving (i.e., Evaluate the solution and Find alternative solutions). As can be seen from a comparison of the two processes, the computational thinking process basically aligns with the problem-solving process, but focuses more on students' abstract thinking, which is the most significant difference between computational thinking and problem-solving (Wing, 2008).

Relationship between computational thinking skill and problem-solving competencies. The quantitative data indicated a significant, strong positive correlation between students' computational thinking skills and their problem-solving competencies (with *rs* = .64, *p* < 0.001). In the qualitative data, this relationship was also found. Those who performed better at solving problems may also be better computational thinkers. This might be because computational thinking involves the problem-solving process, as mentioned before. Also, computational thinking (CT) shares some aspects with problem-solving competencies, such as problem-solving skills (i.e., problem definition and formulation, generation of alternative solutions, decision making, solution implementation, and verification) and the problem-solving process, because "computational thinking involves solving problems" (J. M. Wing, 2006, p. 33) and "CT's essence is thinking like a computer scientist when confronted with a problem" (Grover & Pea, 2010). Wing (2008) and Yadav et al. (2011) also pointed out that encouraging students' computational thinking might potentially enhance students' problem-solving skills.

6.2 RQ2: What Are the Changes in Students' Problem-Solving Competencies and Computational Thinking Skills in a Robotics Learning Environment?

To answer this research question, both the quantitative data (i.e., from the pre-/post-tests and rubric scoring) and the qualitative data (i.e., from interviews and classroom observation) were analysed. The findings showed that students' PS competencies and CT skills were significantly enhanced during the summer camp of this research.

6.2.1 Changes in Students' Problem-Solving Competencies

By using CSPSI, which was validated in a pilot study, the current study conducted pre-/post-tests to measure students' problem-solving competencies before and after the summer camp. Both students' perspectives (students' self-reports) and the teacher's perspective (the teacher's evaluation of every student) were counted for triangulation. Significant positive correlations were found between students' perspectives and the teacher's perspective in the pre-test; meanwhile, in the post-test, not all the results showed this correlation. Also, interestingly, students scored themselves higher than the teacher did in both the pre- and post-tests, which reminds us that students might overestimate their own competencies and that the teacher might underestimate students' competencies. It is reasonable that the perspectives of the students and the teacher vary, as how students perceive their own abilities is quite different from the perception of the teacher. Through only numeric data, it is difficult for us to know which side is "fact," especially when the results suggested acceptable reliability coefficients ($\alpha > .70$) on both sides. Therefore, this research not only provided students' scores and the teacher's scores respectively, but also reported the merged scores of the two groups to address the problem of information bias. Additionally, Wilcoxon tests were conducted for each group of data (i.e., student, teacher, and merged data). As summarized in

Table 16, the results indicated that students' problem-solving competencies significantly increased, with a large effect size and power, regardless of the data group.

Table 66

Summary of Results from Wilcoxon Tests (N = 32)

	Student	Teacher	Merged
Pre-test	9.545	8.598	9.071
Post-test	10.469	10.018	10.243
Wilcoxon test	$z = -3.210$ $p = .001$	$z = -4.905$ $p < .001$	$z = -4.625$ $p < .001$
Effect size	$d = .692$	$d =2.015$	$d = 1.362$
Power analysis	.997	1	1

To triangulate the quantitative data, this research also analysed qualitative data from semi-structured interviews and classroom observation. Qualitative findings corroborated the quantitative results and provided more details and evidence of the development of students' PS competencies. Therefore, for RQ2, both quantitative results and qualitative findings of this research concluded that students' problem-solving competencies in a robotics learning environment had improved. This conclusion echoes those of previous studies that identified students' problem-solving enhancement in robotics education (Atmatzidou et al., 2018; Castledine & Chalmers, 2011; Liu et al., 2013).

Moreover, the findings agreed with the problem-solving model proposed by Chang et al. (2004), which included problem-solving self-efficacy (PSSE) and problem-solving skills (PSS). Another widely-used model for measuring problem-solving competencies was

Heppner and Petersen’s (1982) PSI, which was based on D’Zurilla and Goldfried’s problem-solving theories in 1971. However, in 1995, D’Zurilla and Maydeu-Olivare offered a criticism, saying that the structure of PSI lacked connections to problem-solving theories. Therefore, Maydeu-Olivares and D’Zurilla (1997) revised the PSI to make it a more theoretical form. After critically reviewing and comparing Heppner’s model and D’Zurilla’s model, this research adopted the latter one to evaluate students’ problem-solving, as D’Zurilla’s model could explicitly explain students’ problem-solving in robotics education. According to the definition in Chang et al. (2004), problem-solving self-efficacy means “belie[f] in one’s personal ability to solve problems successfully” (p. 15), while problem-solving skill refers to “the cognitive and behavioural activities by which a person attempts to understand problems and find effective solutions or ways of coping with them” (p. 14). Specifically, four problem-solving skills were included: problem definition and formulation, generation of alternative solutions, decision making, and solution implementation and verification (Chang et al., 2004; Maydeu-olivares & D’Zurilla, 1997). This research suggests that D’Zurilla’s model is suitable for evaluating students’ problem-solving competencies in a robotics learning setting, while few previous studies adopted this model in robotics education.

Additionally, this research tried to connect the problem-solving process to problem-solving competencies, to obtain a better understanding of the stage(s) in which students’ competencies/skills might be mainly involved. The findings showed that: a) In the stage of identifying the problem, students’ problem definition and formulation skills were involved; b)

In the stage of coming up with solution(s), the skill of generation of alternative solutions was involved; c) In the stage of making a decision, the decision making skill was involved; d) In the stages of carrying out the solution and evaluating the solution, the skill of solution implementation and verification was involved; and e) In the stages of coming up with solution(s) and evaluating the solution, problem-solving self-efficacy was involved. Figure 11 summarizes the above conclusions and provides a clear framework connecting the problem-solving process and competencies. This framework explicitly demonstrates the relationship between students' problem-solving process and competencies, as previous studies rarely discussed this connection. It is meaningful because it allows us to determine in which stage(s) we can facilitate certain competencies/skills in students.

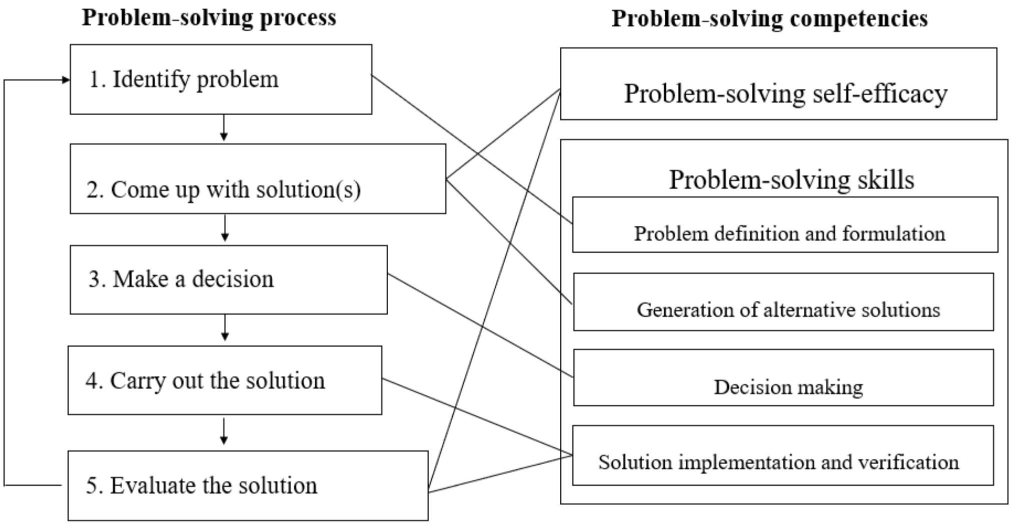

Problem-solving process
1. Identify problem
2. Come up with solution(s)
3. Make a decision
4. Carry out the solution
5. Evaluate the solution
Problem-solving competencies
Problem-solving self-efficacy
Problem-solving skills
Problem definition and formulation
Generation of alternative solutions
Decision making
Solution implementation and verification

Figure 11. Relationship between Problem-Solving Process and Competencies.

6.2.2 Changes in Students' Computational Thinking Skills

The current study scored every students' daily performances by using a computational thinking rubric. Both the teacher's perception and the researcher's perception were counted to address the triangulation matter. The results of Cohen's Kappa test indicated that the two raters' measures were consistent ($.54 < \kappa < .92$, $p < .001$). This was because the researcher communicated well with the teacher about the scoring criteria before practicing, so even though the scoring was conducted independently, the two raters showed a high amount of agreement regarding students' performances. Also, it is reasonable to merge two raters' scores. No one-side data were presented, as the scores were from two external raters, which is different from the situation in measuring students' problem-solving competencies (i.e., scores were from students [subjective raters] and the teacher [external rater]).

Merged data were analysed by Wilcoxon tests to compare students' first-six-days performances of CT to their last-six-days performances. The results of the Wilcoxon tests indicated that students' CT skills significantly increased from the first six days ($M_{\text{first}} = 7.33$, $Mdn_{\text{first}} = 7$) to the last six days ($M_{\text{last}} = 9.39$, $Mdn_{\text{last}} = 8.96$), with $z = -4.94$, $T = 528$, and $p < .001$.

Additionally, a gradual increasing trending was found through day-by-day data. However, students' CT skill did not reach the moderate level until the end of the summer camp. This might be because the duration of the summer camp was for only one month and participants were too young to reach a high level of CT during such a short-term project.

More details about six components of CT were found in the trending lines. Students' logical thinking was enhanced most significantly and surpassed the moderate level on the last day. This suggests that, during the robotics summer camp, students' programming process was increasingly logical to a higher level, which echoes the results of previous studies (Eguchi, 2014; Leonard et al., 2016). Students' progress in terms of their abilities for abstraction and analysing and implementing solutions was also remarkable. Because abstraction is the essence and foundation of computational thinking (Bers, 2010; Wing, 2006), students' remarkable progress in abstraction further verified their improvement in CT skill. Analysing and implementing a solution was also deemed to be a debugging skill, which is a crucial skill for real-world programmers (Leonard et al., 2016); students' enhancement of this skill revealed that they programmed in an increasingly professional manner. However, students' abilities for formulating problems and generalizing and problem transfer were limitedly enhanced. Eguchi (2012) mentioned that, for young-aged children (i.e., Grades 5 to 7 in Eguchi's study), formulating problems is difficult because children have trouble distinguishing between problems, tasks, and troubles. This research also found that, for students in grades 1-3, formulating problems was challenging, though with the teacher's support, this situation gradually improved. The important role of S-T interactions in CT development will be discussed later. As for students' algorithm using, interestingly, the starting level was higher than that of other CT components, while the trending increased slightly. This might be because, during the first-week teaching, algorithm issues were demonstrated explicitly, especially in the first class, and lesson tasks were relatively easy, so that students could use the algorithm properly. When the complexity of the lesson tasks increased, students had trouble using the proper algorithm.

To triangulate the quantitative data, this research also analysed the qualitative data regarding students' CT performances, obtained from semi-structured interviews and classroom observation. Because CTR was based on ISTE's (2014) model of computational thinking, in qualitive data analysis, this was used to provide evidence of students' computational thinking development. Qualitative findings corroborated the quantitative results and provided more details and evidence of students' enhancements in computational thinking skill. The findings suggested that ISTE's model is suitable for evaluating students' CT skills in a robotics learning environment.

Moreover, this research tried to connect the CT process to CT skills, so as to better understand in which stage(s) students' skills might be mainly involved. Specifically, students' abilities for formulating problems, abstraction, logical thinking, and algorithm using were mainly involved in the *Abstraction* stage. Meanwhile, in the *Automation* and *Analysis* stages, their abilities for analysing and implementing solutions and generalizing and problem transfer were mainly involved. In Figure 12, the relationship between the CT process and CT skills mentioned above was summarized. This framework allowed teachers and researchers to determine the focused CT skills in specific stages of CT.

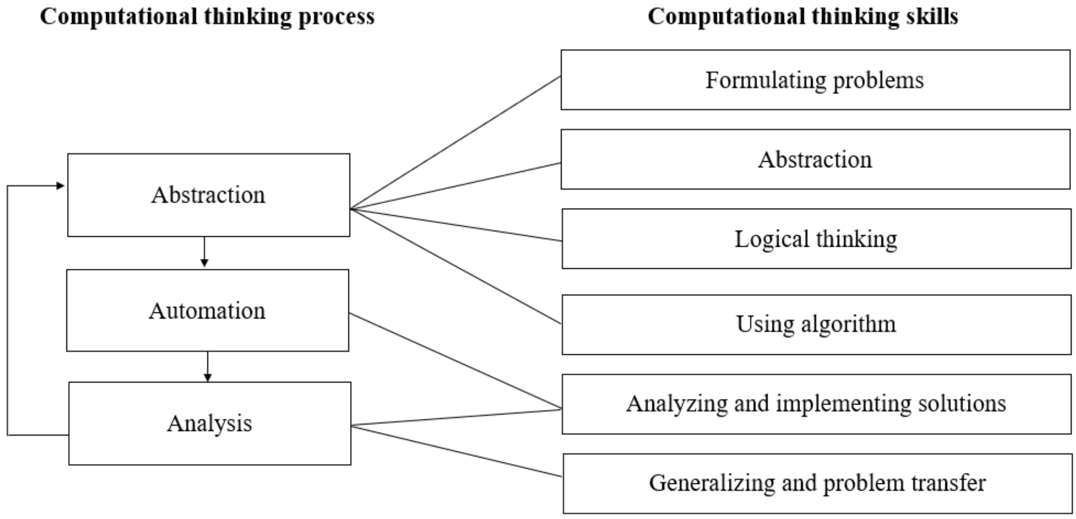

Computational thinking process
Computational thinking skills
Abstraction
Automation
Analysis
Formulating problems
Abstraction
Logical thinking
Using algorithm
Analyzing and implementing solutions
Generalizing and problem transfer

Figure 12. Relationship between CT Process and Skills.

6.3 RQ3: What Are the Relationships between Three Classroom Interactions (i.e., S-T, S-S, and S-R Interactions) and Students' Learning Outcome Development (i.e., PS Competencies and CT Skills)?

Although students' learning outcome measures increased significantly during the summer camp, we should further explain why and how students' problem-solving competencies and computational thinking skills developed. Therefore, within the scope of classroom interactions, this research focused on the relationships between three classroom interactions (i.e., S-T, S-S, and S-R interactions) and students' learning outcome development (i.e., problem-solving competencies and computational thinking skills). Two dimensions of classroom interactions were concerned. On the one hand, in quantitative analysis, students' time spent on three classroom interactions was counted and tested by Spearman's correlation analysis with students' learning outcome enhancements (i.e., the mean difference of pre-

/post-test measures for PS and of the first-six-days/last-six-days scores for CT). On the other hand, in qualitative analysis, by tracking focal samples' performances, interactions that might be effective in students' learning outcome development were characterized. Significant results and findings will be discussed as follows.

6.3.1 Correlation Between Time Spent on Three Class Interactions and Learning Outcome Enhancement

During the summer camp, which lasted for a total of 1080 minutes, the mean time spent on Student-Teacher interactions was about 289 minutes (25%)—the lowest amount of classroom interactions. This result is reasonable because S-T interaction took place mainly during each class's beginning section, when the teacher introduced the lesson task and taught the students how to build and program. After this section, when students were building and programming, the teacher coached and scaffolded them individually. Because all the students expected to receive the teacher's support, the time spent on each student might be limited. The correlation between time spent on S-T interaction and the enhancement of students' PS competencies was significantly positive (*rs*(30) = .42, *p* = .015), while only the PSS subscale showed significant results. In other words, those students who spent more time on S-T interactions performed better in problem-solving skills, but not in problem-solving self-efficacy. This might be because those students who interacted more with the teacher could also have higher chances of learning PS skills from the teacher. In terms of students' CT skills, there was a significantly positive correlation between time spent on S-T interactions and students' CT skills enhancement (*rs*(30) = .60, *p* < .001), while only two CT skills (i.e., Abstraction and Analysing and implementing solutions) showed similar correlations. These two CT skills are difficult for young children to master because abstraction and debugging (i.e., analysing and implementing solutions) are rather professional, even for programmers. Hence, students who interacted more with the teacher—an "expert" in programming—might gain more opportunities for learning these two professional skills. Additionally, the results above remind us about the important role of S-T interaction in cultivating students' PS and CT skills, although for each student the time spent on S-T interactions was limited.

The greatest amount of time—about 484 minutes (41%)—was spent on Student-Student interactions. This was because "the robotics summer camp is more open and has a lot of room for students to interact with others," as explained by the teacher. What should be noted is that, although no group/pair-formed activity was involved in the summer camp, S-S interactions still existed. Much previous evidence identified that group collaborative interactions might help promote students' problem-solving development (Collins et al., 1991; Järvelä, 1999; Larkins et al., 2013; Mitnik et al., 2009). Meanwhile, in this study, the teacher focused more on young children's natural instincts to create their own special robot. Additionally, in a previous study of Yuen et al. (2014), students working in a group had to wait for their turn to use robot kits or computers; therefore, children who were waiting could have been engaging in off-task activities such as chatting and playing with others.

Considering the reasons above, this study allowed children to work individually and to handle their own robots and computers, but it also encouraged interactions between or among students. Both time spent counting and classroom observation in this study suggested that, though the students worked individually, they still tended to interact with peers more than with the teacher and that they could also benefit from S-S interactions. The results of the correlation analysis indicated that time spent on S-S interaction was positively correlated to students’ problem-solving self-efficacy (*rs*(30) = .55, *p* = .001) but non-significantly correlated to problem-solving skills. This might be because, in S-S interactions, there is no authority, so spending more time on this kind of equal interaction allowed students to build their own confidence with less pressure. Regarding the CT skills, enhancement was positively correlated with time spent on S-S interaction (*rs*(30) = .64, *p* < .001). Specifically, for abstraction, logical thinking, and analysing and implementing solutions, positive correlations with S-S interaction were found. Students can learn not only from the teacher but also from their peers. While the time available for S-T interactions is usually limited, peer interactions could be a good supplement. Therefore, encouraging peer interactions in robotics learning is necessary for promoting students’ CT skills.

Student-Robot interactions took up about 397 minutes (34%) during the summer camp. As reviewed before, Student-Robot interaction was rarely discussed in RE studies. However, the significance of this kind of interaction in RE should receive more attention (Jung & Won, 2018). Therefore, this research highlighted the importance of S-R interaction in RE. Students’ time spent on S-R interactions was positively correlated with their problem-solving competencies (*rs*(30) = .46, *p* = .008). Specifically, the significant result was found in the PSS subscale but not in the PSSE subscale. Regarding CT skills, although no significant result was found for each skill, the total score of CT skills was positively correlated with time spent on S-R interactions at a significant level (*rs*(30) = .51, *p* = .003). The reason for the results above might be that interacting with a robot allowed students to practice skills tangibly and visibly, which promoted their PS and CT skills.

All correlations between three classroom interactions and students' learning outcome enhancements were summarized in Figure 13. What should be noted is that the three interactions could possibly overlap—for example, students interacted with the robot while interacting with their peers. Additionally, discussion, for now, was based only on quantitative data analysis, however, correlations found between classroom interactions and students' learning outcomes should not be taken as causal relationships. Also, time spent is just one side of three interactions; the kinds of interactions that might be effective in determining students' learning outcome development should be further specified. Therefore, qualitative data from the learning process were also explored and will be discussed later.

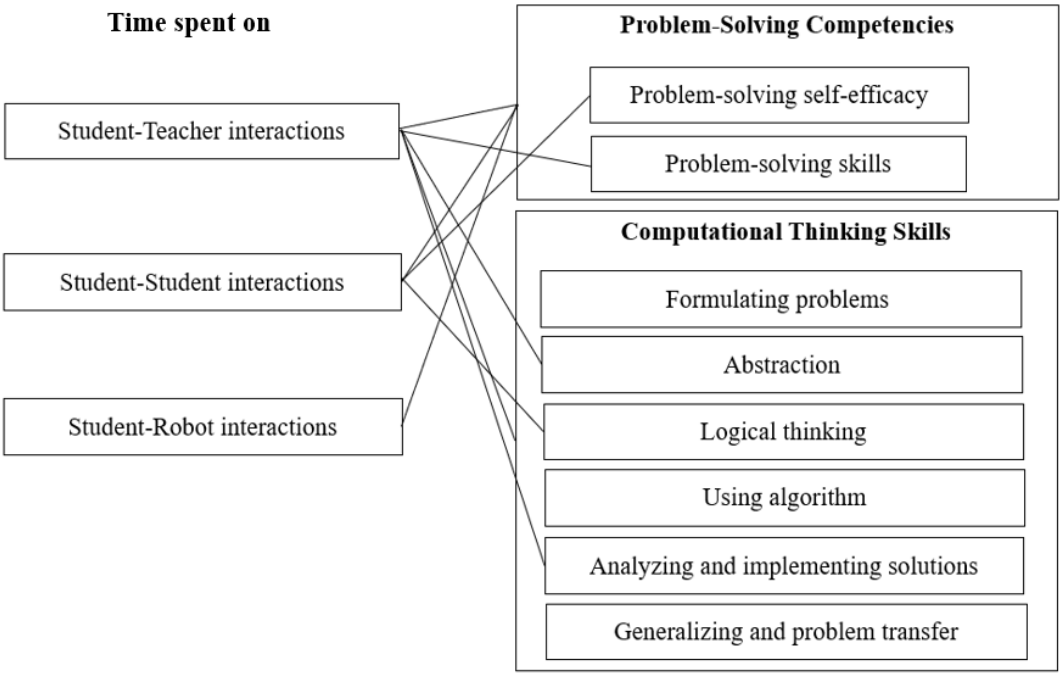

Time spent on
Student-Teacher interactions
Student-Student interactions
Student-Robot interactions
Problem-Solving Competencies
Problem-solving self-efficacy
Problem-solving skills
Computational Thinking Skills
Formulating problems
Abstraction
Logical thinking
Using algorithm
Analyzing and implementing solutions
Generalizing and problem transfer

Figure 13. Summary of Correlations between Time Spent on Three Classroom Interactions and Students' Learning Outcome Enhancements.

6.3.2 Effective S-T Interactions in Students' Learning Outcome Development

For the development of both problem-solving competencies and computational thinking skills, cognitive apprenticeship teaching methods (i.e., including modelling, coaching, scaffolding, articulation, reflection, and exploration) and students' active engagement in these teaching methods, were found to be effective S-T interactions.

Effective S-T interactions in developing problem-solving competencies. In previous studies, cognitive apprenticeship teaching was deemed to be a useful teaching model for supporting students' problem-solving development in S-T interactions (Collins et al., 1991;

Järvelä, 1999). Larkins and colleagues (2013) attempted to apply cognitive apprenticeship teaching in robotics education; however, they did not measure students' problem-solving performance. The current study tried to apply cognitive apprenticeship teaching to the robotics summer camp and focused on its effectiveness in students' problem-solving development. The findings suggested that cognitive apprenticeship teaching in robotics education helps develop students' problem-solving competencies. This might be because the principle of apprenticeship teaching is to make thinking visible (Collins et al., 1991). Through all those apprenticeship teaching activities, students learned tacit knowledge involving problem-solving heuristics and "the strategies that control the problem-solving process" (Collins et al., 1991, p. 12). Consequently, through the teacher's modelling, coaching, scaffolding, and encouraging articulation, reflection and exploration, on every day of the summer camp, students were gradually guided to think like experts.

By modelling, including demonstrating the task and techniques involved in the task, as well as possible problems that might occur, in visible ways (such as drawing on the blackboard, using a sample vehicle, and handling the robot), the students were allowed to observe and generate "a conceptual model of the processes that are required to accomplish it" (Collins et al., 1991, p. 13). By externalizing the implicit processes, students can observe and understand the rationale behind the problem-solving process (Larkins et al., 2013). Also, the modelling activities may be especially effective in the skill development of *Problem definition and formulation*. This is because, for young-aged children, identifying or defining problems is a difficult task (Eguchi, 2012). Therefore, they need an expert to model a real problem.

Coaching served the students when they raised specific problems or during critical moments (Collins et al., 1991; Larkins et al., 2013). The coaching method, for students, was especially effective in promoting students' development of the skill of *Generation of alternative solutions*. Because young students often use intuition to generate solutions, as stated previously, the teacher's coaching can help them cognitively understand the tacit process of generating a solution.

In a robotics learning environment, there are many resources for scaffolding, such as the robot kits, the internal-test model of Intelligent Brick, the sample programs of Scratch, and so on. The timing of scaffolding is important. The teacher should determine the students' current skill level and provide appropriate assistance for students to achieve higher-level skills (Collins et al., 1991). In addition, "as the student step-by-step performs better, the teacher gradually fades his or her helping process" (Järvelä, 1999, p.253). Scaffolding shares some similarities with coaching: Both take place when students carry out the task; they support students' problem-solving process when they seek help; they are especially effective in the skill development of *Generation of alternative solution*; and they might be helpful for students' problem-solving self-efficacy. In fact, these two methods were often discussed together (Järvelä, 1999). However, the main difference between these two methods might be that while scaffolding is one method of coaching, it is not the only method. Through other ways of coaching, such as offering hints, modelling, and feedback, the teacher could also coach students to perform better.

Articulation is intended to guide students in making their tacit knowledge, internal reasoning, or problem-solving process concrete and explicit (Collins et al., 1991; Järvelä, 1999; Larkins et al., 2013). In articulation, through systematic questioning, the teacher encourages students to analyse the reason behind a problem and guided them to articulate the thoughts behind the solutions they generated and their decisions. Therefore, articulation is especially effective in the skill development of *Generation of alternative solutions* and *Decision making*. Also, students' problem-solving self-efficacy might be enhanced by articulation because they have the opportunity to express their thought process.

Reflection allowed students to compare their own performance in the problem-solving and thinking process to that of the "experts" (Järvelä, 1999). By reflecting on their own work, students could recognize their strengths and weaknesses when solving problems. This is effective for all problem-solving skills, as students' reflections were not only on one specific stage of problem-solving. However, we should notice that, requiring students to compare their works to others' might hinder students' self-efficacy development, because students might feel frustrated when they find others' works are better than their own. Therefore, it is important to tell students that reflection is not for comparing but for reviewing.

In this research, the exploration method stimulated students to practice the problem-solving process and develop their own problem-solving modes (Collins et al., 1991; Larkins et al., 2013). Therefore, this method is also effective for all problem-solving skills. Moreover, exploration enhanced students' problem-solving self-efficacy by providing students with the opportunity to face challenges and solve problems on their own.

As discussed above, the cognitive apprenticeship teaching methods applied in this research were found to be effective in promoting students' problem-solving competency development; however, S-T interaction is a mutual process, so students' engagement is also important. Pianta, Hamre, and Allen (2012) pointed out that students' engagement is not only an outcome of teaching but also "a mediator of impacts that teachers have on student outcomes through their interactions with children" (p. 366). Therefore, this research acknowledged that cognitive apprenticeship teaching provided opportunities for students' engagement (Collins et al., 1991), but the question of whether students were actively engaged in cognitive apprenticeship teaching is significant to their problem-solving competencies. In the view of social constructivism, young children develop their thinking abilities through interacting with others and the outside world, and teacher play an important role of scaffolding students to achieve "a continual movement from the current intellectual level to a higher level which more closely approximates the learner's potential" (Amineh & Asl, 2015, p. 14). Therefore, students' active engagement, in no matter S-T interactions or other learning activities, was found to be a positive factor influencing students' learning outcomes (Fredricks, Blumenfeld, & Paris, 2004). The current study found that successful cognitive apprenticeship teaching relies on students' active engagement in S-T interaction, especially in an environment full of problems and challenges (such as a robotics summer camp, in this research).This kind of positive and active S-T interaction may lead students to solve problems more rationally and to gradually become potential constructive problem solvers, as they had more opportunities to learn from the expert (i.e., the teacher) in problem-solving. On the other hand, those who did not benefit from cognitive apprenticeship teaching failed to do so because they were inactively engaged in S-T interaction and, thereby, had fewer opportunities to learn from the expert.

Effective S-T interactions in developing computational thinking skills. As for CT skills development, the teacher's cognitive apprenticeship teaching was also effective. Moreover, students' active engagement in S-T interactions was found to be effective for their CT skill development. Larkins et al. (2013) found that cognitive apprenticeship teaching could develop skills in computational thinking; however, without a clear model of CT skills, they failed to demonstrate how students' CT skills were cultivated by cognitive apprenticeship teaching. Therefore, in this research, how students benefited from cognitive apprenticeship teaching was demonstrated.

In the summer camp, the teacher modelled by orally formulating problems and drawing the flow chart of the programming process. Therefore, modelling is especially effective in developing students' formulating problems and abstraction skills. When students were programming, the teacher observed and coached them by providing suggestions, reminders, and support, so that students' logical thinking and algorithm using skills were developed. By providing sample program(s) and using the internal-test model of Intelligent Brick, the teacher engaged in scaffolding that was effective mainly in developing students' abstraction, logical thinking, and algorithm using skills, as students could visually see how the robot detected the real-world situation, which allowed them to logically abstract the robot's reaction into programming. Because programming problems were usually abstracted, the teacher's articulation may have made students' tacit thinking process more concrete and explicit. When students were debugging their programs, the teacher encouraged them to simulate the robot's routine orally or physically, so that the students could easily find the faulty part of the program. The articulation method is especially effective in developing students' skills of analysing and implementing solutions. Reflection allowed students to compare their computational thinking process to an expert's, so that all CT skills were effectively involved. Exploration allowed students to generalize and transfer the computational thinking process from the given task to more challenging and creative tasks, and thereby effectively develop the skill of generalizing and problem transfer.

Similar to findings for problem-solving competencies, the question of whether students were actively engaged in S-T interaction is significant to their computational thinking skills. Without students' active engagement, the teacher's unidirectional teaching would be ineffective. Students' active engagement may allow the teacher to provide targeted support more accurately, while inactive engagement may hinder the teacher's support or lead to over-reliance on the teacher's guidance. What should be noted is that students' active engagement in S-T interaction means that not only did they spend a lot of time interacting with the teacher, but they also effectively engaged in S-T interactions (see Kevin's example).

To conclude, for the development of both problem-solving competencies and computational thinking skills, cognitive apprenticeship teaching methods (i.e., modelling, coaching, scaffolding, articulation, reflection, and exploration) and students' active engagement in these teaching methods, were found to be effective S-T interactions. How students' learning outcomes were developed was specifically discussed. What should be noted is that, although different teaching methods might have their own targeted abilities/skills, this does not mean that other competencies/skills were not involved at all. For example, modelling may be especially effective in developing the skills of *Problem definition and formulation* and *Abstraction*. However, other problem-solving skills could also benefit from modelling; *Problem definition and formulation* and *Abstraction* were just the two that might benefit most. The purpose of specifying which methods may mainly develop which skills is to better understand how students' learning outcomes were developed.

6.3.3 Effective S-S Interactions in Students' Learning Outcome Development

In the original conceptual framework of this research, S-S interactions contained two kinds of cooperative interactions (i.e., Discussion and Observational learning) and off-task interactions. Analysis of the qualitative data found that discussion and observational learning were effective S-S interactions, while off-task interactions (such as unstructured chatting, playing with the robot or peers, and other interactions unrelated to the lesson task) were ineffective at developing students' PS competencies and CT skills. Regarding the development of PS competencies, *division of work* was added to the conceptual framework, so that the three effective interactions were: a) discussion; b) division of work; and c) observational learning. For CT skill development, unlike the finding in the PS competencies, division of work was not included in the effective S-S interactions for the CT skill, as the CT skill was mainly for solving programming problems, while students tended to program on their own. In addition, because most of the programming tasks at this summer camp had simple structures, labour division was not necessary.

Effective S-S interactions in developing problem-solving competencies. In the original conceptual framework of this research, S-S interactions contained two kinds of cooperative interactions (i.e., Discussion and Observational learning) and off-task interactions (such as unstructured chatting, playing with the robot or peers, and other interactions unrelated to the lesson task). Through qualitative analysis, cooperative interactions were found to be effective S-S interactions, while off-task interactions were ineffective at developing students' PS competencies. In addition to discussion and observational learning, contained in the original framework, division of work was included in the effective S-S interactions. Therefore, in the revised conceptual framework, discussion, division of work, and observational learning were effective S-S interactions for problem-solving development.

The S-S discussion involved idea-sharing and conflicting (Azmitia, 1988; Duckworth, 2005). Through idea-sharing and conflicting, different ideas of solving problems were exchanged (Duckworth, 2005), which led to the “collective creation of knowledge” (Eguchi, 2012, p. 21). As stated by Duckworth (2015), “It was always Piaget’s belief that discussion among peers was the most likely means of moving learners beyond their current understanding” (p. 262). Additionally, discussion allowed students to make problem-solving thinking visible to others and to themselves (Duckworth, 2005). Sometimes, children may prefer to discuss things with peers rather than with the teacher; therefore, peer discussion was a good supplement to the teacher’s articulation.

It is surprising that, even though they were young and had to deal with relatively complex tasks, the children sometimes had ideas regarding the division of work. What should be noted is that, although no grouped/paired activities were involved in this summer camp, the division of work unexpectedly occurred. It might be because the class size was small and the class was relatively free, students can move around and talk freely, so they naturally found ways to cooperate. During this process, some students put forward the idea of division of work, and other students concurred with them, so they came to agreement on the work division. The division of work might be beneficial to students’ problem-solving self-efficacy.

This might be because the “leaders” (e.g., two focal samples, Iris and Adam) gained more confidence by leading others in solving problems; their peers may also be more confident when solving problems because they knew that there were peers on whom they could rely. In addition, work division was effective and time-saving, so they had more time to work out other problems and accomplish the task. This would also enhance their problem-solving self-efficacy and skills.

Azmitia (1988) found that observational learning is effective for problem-solving development because “novels” can observe and learn from “experts.” Commonly, in one classroom, one person might solve problems more productively than others; therefore, he/she becomes an “expert” among peers, and other students can observe how the expert solves problems, as well as learn from the expert’s experiences, thereby gradually becoming “experts” themselves. Through meaningful observation, students’ abilities for generating alternative solutions and decision making were enhanced; thus, the progress from novice-to-expert could be seen. What should be critically considered is that some children preferred to work alone and independently. However, although their verbal interactions with peers were less frequent, some of them might have taken part in observational learning to solve problems; their problem-solving competencies might also be influenced.

It should be noted that these three kinds of interactions were usually overlapped and connected. For example, before and after the division of work, a discussion was necessary. Also, when observing others’ work, students sometimes had discussions with their peers.

Effective S-S interactions in developing CT skills. The original conceptual framework of this research included cooperative interactions (i.e., discussion and observational learning) and off-task interactions. However, an analysis of the qualitative data revealed no sufficient evidence supporting the idea that off-task interactions were effective. Consequently, in this research, discussion and observational learning were found to be effective S-S interactions for developing CT skills.

In S-S discussion, students reflected on the teacher's modelling and logically gathered information from the teacher for use in their own design. Through the discussion process, by thinking aloud to each other, they abstracted the problem more concretely. Also, through discussion, they could recognize in what areas they needed the teacher's help, which meant that the teacher's support could be more accurate. Moreover, in terms of algorithm using, a peer's reflective discussion may have provided useful information about which algorithm logic is more suitable for the lesson task. In analysing and implementing solutions, discussion allowed students to share their debugging process with others. In generalizing and transferring the problem-solving process, discussion may have reminded students about the possibility of using a known problem-solving process to solve new problems.

Observational learning allowed students to compare their own work to that of others; therefore, they learned how to succeed in programming and how to program and debug like "experts." Thus, encouraging students' observational learning might help promote their CT skills. What should be noted is that observing did not mean copying. Observational learning requires students' critical thinking to distinguish between which parts of others' work could be adopted and which parts could not, while copying is to simply accept others' ideas and directly use others' programs. Because copying others' programs is easier than copying others' building design, it is especially important to focus on this situation in the programming part of robotics learning.

To conclude, discussion and observational learning were effective in students' CT skill development. Through discussion, useful information was exchanged among students, and they could think more concretely. Specifically, when formulating problems, discussion stimulated students to break down a general problem into smaller ones. When abstracting problems, discussion helps students connect the real world to the abstracted programming process. Through observational learning, students learned from others' failures and successes and reflected on their own work. Also, observational learning allowed students to debug more effectively. Through discussion and observational learning, students' programming process became increasingly logical and they tended to adopt their algorithm logics more properly.

In summary, for the development of problem-solving competencies, three cooperative S-S interactions were characterized as effective: discussion, division of work, and observational learning. For CT skill development, two cooperative S-S interactions were characterized as effective: discussion and observational learning. How students' learning outcomes were developed through effective S-S interactions was specifically discussed. What should be critically considered is that quantitative analysis showed a nonsignificant correlation between time spent on S-S interactions and problem-solving skill development, while in qualitative analysis, this research suggested that effective S-S interactions could also benefit students' problem-solving skill. This might be because S-S interactions come in various forms, some of which—such as off-task chatting and playing, which have nothing to do with robotics learning—might not be effective for skill development (Yuen et al., 2014).

6.3.4 Effective S-R Interactions in Students' Learning Outcome Development

Previous studies rarely discussed S-R interaction; however, this kind of interaction should be explored because it provides more information about students' natural learning process, which might be meaningful to robotics education practice (Jung & Won, 2018). Some studies regarded the robot as playing a passive role in RE and regarded building and programming as unidirectional behaviours (such as Mubin et al., 2013). However, either building or programming rely on obtaining information from the robot's behaviours, so it is actually a mutual interaction process. Interviews with students revealed that many of them thought that they interacted with their robots. In this research, some effective S-R interactions were characterized. Specifically, for the development of problem-solving competencies, structure refining and program debugging were two effective S-R interactions. For the development of CT skills, selection of inputs, observational investigation, and participatory investigation were found to be effective.

Effective S-R interactions in developing problem-solving competencies. S-R interactions played an important role in the development of students' problem-solving competencies because all problem-solving stages relied on S-R interactions; identifying a problem and evaluating the effectiveness of a solution require observing the robot's behaviours and getting information from the robot's output (i.e., from the Intelligent Brick). When generating a solution or finding alternative solutions, one must explore the possible outcomes of solutions by refining the structure of a robot and debugging with a robot, time after time. Also, one can implement one's solutions only by handling or programming for a robot.

In the original conceptual framework of this research, building, programming, observing, and participatory investigation were included in S-R interactions. However, during the analysis of the qualitative data, it was found that structure refining and program debugging were more suitable for characterizing the effective S-R interactions. This was because, as discussed before, two typical types of problems involved in robotics learning are building problems and programming problems. Therefore, when analysing S-R interactions, it is reasonable to see how students solve these two kinds of problems by interacting with the robot. Also, as compared to "building" and "programming," which describe students' one-way behaviour, “structure refining” and “program debugging” are more suitable for describing interactions. Moreover, in the study of Castledine and Chalmers, (2011), *software programming changes* and *change in robot design* were included as strategies for students’ problem-solving in robotics education. Inspired by these two terms, this research characterized two kinds of S-R interactions that might be effective in students’ problem-solving competencies: structure refining and program debugging.

Structure refining was for solving building problems. When building a robot, students implemented their designing ideas; however, there was often a gap between their ideas and the handmakes. This is where building problems may come from. Structure refining is not only a common method for solving building problems but also a kind of S-R interaction that involves all problem-solving skills (see Adam’s example). Through repetitive interactions with the robot, structure refining allowed students to practice problem-solving skills tangibly. Also, through the structure refining process, the problem-solving process was implemented. Moreover, students gained more confidence from this process.

Program debugging was for solving programming problems. Debugging is an inevitable process of programming; it seeks to find and resolve problems in a program. When debugging, students observed the robot’s behaviours and reactions to determine whether the robot moved as expected. Thereby, they made decisions about refining programs. Debugging involved the processes of gathering information from the robot and inputting information to the robot, which constitute a bidirectional interaction between student and robot. This interaction is effective for developing students' problem-solving competencies because it involves all problem-solving skills (see Iris' example).

Effective S-R interactions in developing problem-solving competencies

S-R interactions played an important role in students' abstraction—the foundation and essence of computational thinking (Wing, 2006, 2008)—because the robot is a tangible and visible agent between the real world and the abstracted computing world (Bers et al., 2014). Interacting with a robot helps students abstract the real-world situation to the programming process, which could benefit students' CT skill development. In the original conceptual framework of this research, building, programming, observing, and participatory investigation were included in S-R interactions. However, an analysis of the qualitative data revealed that building behaviours were not significantly related to students' CT skills. Additionally, compared to "programming," the term "selection of inputs" proposed by Bers (2010) was more suitable for characterizing how a better computational thinker interacts with the robot. As a result, the critical selection of inputs, observational investigation, and participatory investigation were found to be effective for students' CT skill development and were included in the new conceptual framework.

The selection of inputs is a process of "talking" to the robot through "manipulation of variables and computational instructions" (Bers, 2010, p. 3). Computational abstraction relies on the selection of inputs (Bers, 2010), and the selection of inputs is for abstracting the expected movements of the robot to programming. It was found that those who were critically selecting inputs may achieve a higher level of CT skill because the process of critically selecting inputs involved students' abilities for abstraction (such as simulation), logical thinking, and algorithm using (see Kevin's example).

Observational investigation is a process of "listening" to the robot's behaviours. Computational abstraction also relies on the observation of outputs (outcome data) (Bers, 2010). In pure programming, the outputs are visible on the computer interface but are not tangible. Meanwhile, in robotics programming, the outputs can be observed from the robot's behaviours, which are quite tangible, so it is easier for students to observe and analyse. Observational investigation may facilitate students' ability for abstraction and solution implementation and analysis (see Kevin's example). What should be noted is that observational investigation is more than just watching what happens. Some students watched the robot's performance but usually ignored the robot's misbehaviours and did not seriously regarded an unexpected outcome (see Terry's example).

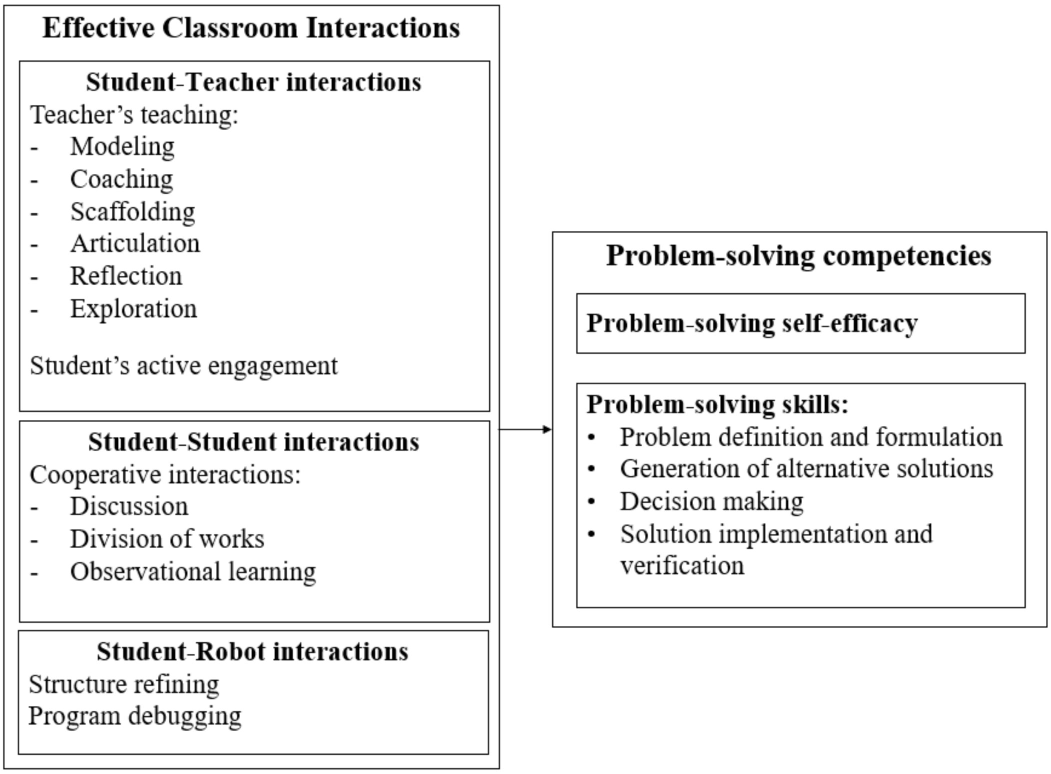
In the study of Levy and Mioduser (2010), *participatory investigation* was a kind of direct bodily S-R interaction regarding programming and debugging, in which "the child's role shifts from designer and observer to that of participant" (p. 28). Participatory investigation allowed students to connect the abstract debugging process to concrete interaction with the robot.

In summary, for the development of problem-solving competencies, structure refining and program debugging were two effective S-R interactions. Structure refining is for solving building problems, while program debugging is for programming problems; both interactions involved all problem-solving skills. For the development of CT skills, the critical selection of inputs, observational investigation, and participatory investigation were found to be effective. These kinds of S-R interactions allowed students to think by doing, by moving from abstracted program designing and debugging to concrete robot behaviours.

6.3.5 Summary: The Revised Conceptual Framework

The discussions above answered the question, “What might be the relationship between three classroom interactions (i.e., S-T, S-S, and S-R interactions) and students’ learning outcome development (i.e., problem-solving competencies and computational thinking skills)?,” which is the focus of this research. Positive correlations between time spent in three types of classroom interactions and two learning outcomes were identified. Also, effective interactions that might develop students’ learning were characterized, and how these interactions develop PS competencies and CT skills was specifically discussed.

Which should be noted is that, when exploring the relationships between classroom interactions and students’ learning outcomes, some qualitative findings were inconsistent with the quantitative findings. For example, the correlation analysis did not support that the time spent on S-T interactions were correlated with students’ PS self-efficacy development, while the qualitative data showed that the effective S-T interactions discussed above could also promote students' PS self-efficacy. However, the author do not attempt to assert that these findings are contradictory, because we should remember that *the time spent on classroom interactions* is quite different from *effective classroom interactions*. The quantitative analysis could help us make some basic judgements, but it might simplify the actual situation in the classroom. This is why this research focused more on the qualitative parts, especially when classroom interactions were too complex to be represented by the length of time. As a result, this research revised the original conceptual framework, to summarize those effective classroom interactions found in this research (see Figure 14).



Effective Classroom Interactions

Student-Teacher interactions

Teacher's teaching:

- Modeling
- Coaching
- Scaffolding
- Articulation
- Reflection
- Exploration

Student's active engagement

Student-Student interactions

Cooperative interactions:

- Discussion
- Division of works
- Observational learning

Student-Robot interactions

Structure refining

Program debugging

Problem-solving competencies

Problem-solving self-efficacy

Problem-solving skills:

- Problem definition and formulation
- Generation of alternative solutions
- Decision making
- Solution implementation and verification

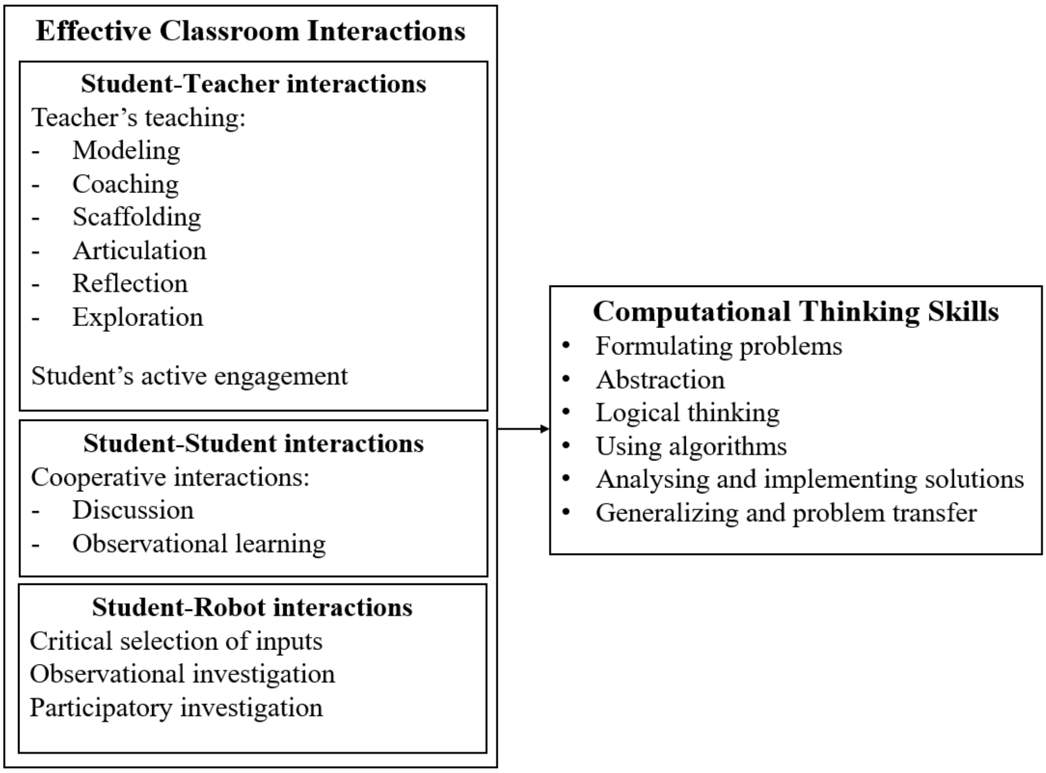

Effective Classroom Interactions
Student-Teacher interactions
Teacher's teaching:
- Modeling
- Coaching
- Scaffolding
- Articulation
- Reflection
- Exploration
Student's active engagement
Student-Student interactions
Cooperative interactions:
- Discussion
- Observational learning
Student-Robot interactions
Critical selection of inputs
Observational investigation
Participatory investigation
Computational Thinking Skills
• Formulating problems
• Abstraction
• Logical thinking
• Using algorithms
• Analysing and implementing solutions
• Generalizing and problem transfer

Figure 14. Revised Conceptual Framework of This Research.

The main revisions of this framework were as follows. Firstly, for both PS competencies and CT skills, students' inactive engagement in S-T interactions and off-task interactions in S-S interaction were removed, as no evidence supported the notion that these interactions were effective. Secondly, for PS competencies, students' division of work was added to the effective S-S interactions, while the four S-R interactions in the original framework (i.e., programming, building, observing, participatory investigation) were replaced by structure refining and program debugging. Thirdly, for CT skills, the four S-R interactions in the original framework were replaced by the critical selection of inputs, observational investigation, and participatory investigation. Lastly, the line connecting classroom interactions and students’ learning outcomes was replaced by arrows, which refers to the positive influences of effective classroom interactions on students’ PS competencies and CT skills.

6.4 Contributions and Implications

As reviewed by Jung and Won (2018), studies on young children’s learning process in RE were insufficient; however:

> “The children’s process-focused studies were very practical in that the findings can suggest more responsive and accessible pedagogical implications for teaching young children. Because the process-focused studies shared detailed processes of children’s robotics learning, the process-focused studies can shed light on authentic aspects of robotics education. In addition, they can provide pedagogical implications relevant to young children’s unique interests, tendencies, and needs.” (p. 10)

The current study is a process-focused study concerning, and providing details about, classroom interactions at a robotics summer camp. Several contributions and implications for RE teaching and research could be concluded as follows.

6.4.1 Implications for RE Teaching

Some conclusions of this research might have implications for future RE teaching. Firstly, the RE teacher should recognize the difference between building problems and programming problems, to better support students' problem-solving and computational thinking development. When facilitating students' problem-solving competencies, the teacher should help students identify problems with two aspects: programming and building. When cultivating students' computational thinking skills, the teacher should remind students to focus on programming problems, especially algorithm logics.

Secondly, the RE teacher should focus on students' intuitive generation of solutions. On the one hand, the teacher could encourage students to use intuition in order to generate inventive solutions. On the other hand, the teacher should help them think aloud for purposes of transferring tacit knowledge (Polanyi, 2009).

Thirdly, the process of problem-solving and computational thinking (i.e., Figure 6 and 7) and the relationship between process and competencies/skills (i.e., Figure 11 and 12) demonstrated in this research provide explicit models for teaching these two competencies in a robotics setting.

Fourthly, in pedagogical design, RE teachers could consider promoting effective classroom interactions found in this research, e.g., adopting cognitive apprenticeship teaching in S-R interactions, encouraging or organizing peer discussion, the division of work, observational learning in S-S interactions, and guiding students to effectively interact with the robot.

Fifthly, it is necessary for RE teachers to critically consider using group/paired activities in robotics settings. As implied by Yuen et al. (2014), students preferred to build and program individually; when they worked in a group, they often wasted time on waiting. Although forcing children to collaborate as a group/a pair might enhance their interactions and learning outcomes (Lee et al., 2013; Toh et al., 2016), students' demands to create their own unique robot should not be ignored. Because the robotics learning environment is commonly open enough for interactions, the adoption of compulsive grouping/pairing forms of S-S interaction should be carefully and critically considered.

Sixthly, the RE teacher should promote bidirectional S-R interactions. S-R interaction is not a unidirectional process of simply watching the robot's movements. The teacher should assist students in gathering information from the robot's outputs.

Last but not the least, although this research was situated in an informal extracurricular environment (i.e., a summer camp), all the findings and relevant models of this research could be transferred to the formal school environment, so it might also provide implications for school teachers. For example, schoolteachers could consider the effective classroom interactions identified by this research when developing or enhancing the school-based RE programs. As introduced in the background, the government of China is promoting RE in K-

12 level. There are many public schools exploring the way of using robotics to cultivate students' life-long learning skills. Thereby, this research might contribute to it.

6.4.2 Contributions to and Implications for RE Research

For RE researchers, some findings of this research could be applied to future studies. Firstly, this research adopted and verified D'Zurilla's model of problem-solving theory (Chang et al., 2004; Maydeu-olivares & D'Zurilla, 1997) in the RE setting, and thereby contributes to relevant theories of problem-solving by connecting the problem-solving process and effective classroom interactions to this model. This model interpreted more detailed process-oriented evidence. Also, based on this model, this research validated a tool, CSPSI, for measuring students' problem-solving competencies in the RE setting in the context of mainland China. Both the model and the tool could be employed in future studies for exploring students' problem-solving in the RE setting.

Secondly, this research adopted ISTE's (2014) model of computational thinking; guided by this model, this research offered more detailed students' process-oriented evidence about CT skills. Future studies could compare the existing models of CT skill in the RE setting (such as Atmatzidou & Demetriadis, 2016; Eguchi, 2014; Lee et al., 2011), as the CT theory is still under-researched (Ioannou & Makridou, 2018). Based on ISTE's model, Leonard et al. (2016) developed a tool, CTR, for scoring students' CT skills. This research adapted this tool to better describe students' CT performances at different levels, in the RE setting. Future studies could further probe this tool in the RE setting, as the assessment of CT skill is still an under-researched area, while CT skill is gaining an increasing amount of attention from RE researchers (Ioannou & Makridou, 2018).

Thirdly, this research provides a view of classroom interactions, which contributes to the understanding of students' learning process in RE. Effective classroom interactions found in this research contribute to pedagogical concerns in the RE area. For S-T interactions in RE, Collins et al.'s (1991) cognitive apprenticeship teaching model was found to be effective for developing both problem-solving competencies and CT skills. This research enriches this model through specifying how students' competencies/skills were developed by this model. Future studies could further explore the application of this model in RE. Also, for S-S interactions in RE, because this research suggested that group/paired-formed activities might ignore students' demands to create their own unique robot, future studies could focus on the problem of balancing students' demands and group/paired-formed activities. Moreover, this research added a new term, "Student-Robot interactions," to robotics classroom interactions; future studies could provide more evidence of S-R interactions to enrich this term.

6.5 Limitations and Suggestions

Several limitations of this research should be acknowledged. Firstly, only 32 valid samples were analysed. In quantitative analysis, a small sample size might lead to statistical problems, although this research tried to provide information about effect size and power analysis to address this limitation. Due to the small sample size, some statistical analyses, such as regression analysis, path analysis, confirmatory factor analysis, and so forth, could not be conducted. Secondly, the summer camp of this research lasted only four weeks. Although short-term research is common in RE research (Toh et al., 2016), long-term research is still necessary for understanding students' learning development. Thirdly, the summer camp of this research did not involve group/pair activities because of the previously mentioned rationales; therefore, this research did not provide information on students' group work. Fourthly, this research was conducted in an informal extracurricular environment because RE is not yet included in the formal curriculum of mainland China. Fifthly, this research focused on two learning outcomes (i.e., problem-solving competencies and CT skills), while other 21st-century skills should also be concerned. Fifth, this research should have adopted a peer examination process for increasing the trustworthiness of the findings, such as when counting students' interactions, although it was not easy to share the data with others given that the size of database is quite huge and the majority of participants are children whose privacy should be strictly guarded.

There are some suggestions for future studies in RE. Firstly, long-term research with a larger sample size, especially in formal learning environments, should be conducted. Secondly, involving grouped/paired-form activities in RE should be critically considered. Thirdly, other 21st-century skills should be assessed and probed. Fourth, though the current study did not measure students' self-efficacy of CT, probing in this aspect of CT could be a significant topic in future RE studies.

6.6 Conclusions

With the deepening discussion of RE, researchers should pay attention not only to the learning outcomes of students, but also to the learning process of students (Jung & Won, 2018). By reviewing relevant literature in the RE field, this research demonstrated the current status of RE research and proposed a conceptual framework. Beyond that, it aimed to explore the features of students' PS and CT in robotics learning, evaluate students' PS competencies together with CT skills in robotics learning, and identify the relationship between classroom interactions and students' PS competencies and CT skills. The mixed-method research was conducted at a four-week RE summer camp in the mainland China, among which the methods of pre-/post-test, rubric scoring, classroom observation, and semi-structured interviews were adopted.

When probing the first research question, "What are the features of students' problem-solving and computational thinking in a robotics learning environment?" the data from the classroom observation and the semi-structured interviews with students and teacher were analysed. Three features of students' problem-solving were found. First, problems that should be solved in robotics learning environments were mainly of two forms: building problems and programming problems. Second, children's PS process generally followed a five-stage cycle: identify the problem, come up with a solution, carry out the solution, make a decision, and evaluate the solution. Third, children's solution-generation process was usually intuitive. Regarding to the features of students' computational thinking, three themes were found. First, CT skill focused on solving programming problems, including conditional logic, iterative logic, and parallel logic. Second, children's CT process generally followed a "three-As" circled process, including abstraction, automation, and analysis. Third, CT skill was related to the problem-solving competencies.

For answering the second research question (i.e. what are the potential changes in students' PS competencies and CT skills in a robotics learning environment?), both qualitative and quantitative data were analysed. For measuring students' PS competencies, the current research used CSPSI, which had been validated in a pilot study, and performed the pre-/post-tests before and after the summer camp. As indicated by the results, students' problem-solving competencies had significantly increased. In order to triangulate the quantitative data, this research also analysed qualitative data from semi-structured interviews and classroom observation. In short, qualitative findings not only corroborated the quantitative results but also provided more details and evidence of the development of students' PS competencies. Therefore, both quantitative results and qualitative findings of this research concluded that students' problem-solving competencies in a robotics learning environment had improved. Additionally, this research tried to connect the problem-solving process to the problem-solving competencies, so as to gain a better understanding of the stage(s) in which students' competencies/skills might be mainly involved. For exploring the change in students' CT skills, the current research scored every students' daily performances by using a computational thinking rubric. Furthermore, the difference between first-six-day measures and last-six-day measures indicated that student's CT skills, including problem formulating, abstraction, logical thinking, algorithm using, analysis and implementation of solutions, and generalizing and problem transfer, significantly increased. Moreover, a gradual increasing trending was found through day-by-day data. However, students' CT skill failed to reach the moderate level until the end of the summer camp. In order to triangulate the quantitative data, this research also examined the qualitative data regarding students' performances, which were obtained from semi-structured interviews and classroom observation. Overall, the qualitative findings corroborated the quantitative results and offered more details and evidence of students' enhancements in CT skill.

The third research question - what might be the relationship between three classroom interactions and students' learning outcome development? - is the focus of this research. In the quantitative analysis, students' time spent on three classroom interactions was counted and tested by Spearman's correlation analysis with students' learning outcome enhancements (i.e. the mean difference of pre-/post-test measures for PS and of the first-six-day/last-six-days score for CT). Consequently, the positive correlations between the time spent in three types of classroom interactions and two learning outcomes were identified. In the qualitative analysis, the interactions that might be effective in students' learning outcome development were characterized by tracking focal samples' performances. Moreover, how these interactions developed PS competencies and CT skills was specifically investigated.

Based on the findings above, some discussions were conducted, and a new conceptual framework was also proposed. Moreover, the contributions and implications of this research were presented. Finally, the limitations of this research and suggestions to future studies were explored.

References

AÇIŞLI, S. (2016). Investigation of the effect of robotic applications in elementary education. In *The Eurasia Proceedings of Educational and Social Sciences* (pp. 391–394). Retrieved from https://dergipark.org.tr/tr/pub/epess/issue/30322/334166

Ackermann, E. (2001). Piaget’s constructivism, Papert’s constructionism: What’s the difference? *Future of Learning Group Publication*, *5*(3), 438–448.

Ackermann, E. (2004). Constructing knowledge and transforming the world. In M. Tokoro & L. Steels (Eds.), *A learning zone of one’s own: Sharing representations and flow in collaborative learning environments* (pp. 15–37). Washington, DC: IOS Press.

Amineh, R. J., & Asl, H. D. (2015). Review of constructivism and social constructivism. *Journal of Social Sciences, Literature and Languages*, *1*(1), 9–16.

Anney, V. N. (2014). Ensuring the quality of the findings of qualitative research: Looking at trustworthiness criteria. *Journal of Emerging Trends in Educational Research and Policy Studies*, *5*(2), 272–281.

Anwar, S., Bascou, N. A., Menekse, M., & Kardgar, A. (2019). A systematic review of studies on educational robotics. *Journal of Pre-College Engineering Education Research*, *9*(2), 19–42. Retrieved from https://doi.org/10.7771/2157-9288.1223

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students’ computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*(B), 661–670.

Atmatzidou, S., Demetriadis, S., & Nika, P. (2018). How does the degree of guidance support students’ metacognitive and problem solving skills in educational robotics? *Journal of Science Education and Technology*, *27*(1), 70–85. https://doi.org/10.1007/s10956-017-9709-x

Azmitia, M. (1988). Peer interaction and problem solving: When are two heads better than One? *Child Development*, *59*(1), 87–96.

Barak, M., & Assal, M. (2018). Robotics and STEM learning: Students’ achievements in assignments according to the P3 Task Taxonomy - practice, problem solving, and projects. *International Journal of Technology and Design Education*, *28*(1), 121–144.

Barak, M., & Zadok, Y. (2009). Robotics projects and learning concepts in science, technology and problem solving. *International Journal of Technology and Design Education*, *19*(3), 289–307. https://doi.org/10.1007/s10798-007-9043-3

Barker, B. S., & Ansorge, J. (2007). Robotics as means to increase achievement scores in an informal learning environment. *Journal of Research on Technology in Education*, *39*(3), 229–243. https://doi.org/10.1080/15391523.2007.10782481

Bascou, N. A., & Menekse, M. (2016). Robotics in K-12 formal and informal learning environments: A review of literature. In *2016 ASEE Annual Conference & Exposition*. New Orleans, Louisiana. https://doi.org/doi: 10.18260/p.26119

Belpaeme, T., Baxter, P., Greeff, J., Kennedy, J., Read, R., Looije, R., … Coti, M. (2013). Child-Robot interaction: perspectives and challenges. In G. Herrmann, M. J. Pearson, A.

Lenz, P. Bremner, A. Spiers, & U. Leonards (Eds.), *International Conference on Social Robotics* (pp. 452–459). Cham, Switzerland: Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-02675-6_45#citeas

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers and Education*, *58*(3), 978–988. https://doi.org/10.1016/j.compedu.2011.10.006

Bers, M. U. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York: Teachers College Press.

Bers, M. U. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research and Practice, 12*(2), 1–20.

Bers, M. U., Flannery, L., Kazakoff, R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, *72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Carbonaro, M., Rex, M., & Chambers, J. (2004). Using LEGO robotics in a project-based learning environment. *Interactive Multimedia Electronic Journal of Computer Enhanced Learning*, *6*(1), 1–19.

Castledine, A. R., & Chalmers, C. (2011). LEGO robotics: An authentic problem solving tool? *Design and Technology Education: An International Journal*, *16*(3), 19–27.

Chambers, J. M., Carbonaro, M., & Murray, H. (2008). Developing conceptual understanding of mechanical advantage through the use of Lego robotic technology. *Australasian Journal of Educational Technology*, *24*(4), 387–401.

Chang, C., Lee, J., Chao, P., Wang, C., & Chen, G. (2010). Exploring the possibility of using humanoid robots as instructional tools for teaching a second language in primary School. *Educational Technology & Society*, *13*(2), 13–24. Retrieved from https://www.questia.com/library/journal/1G1-232945833/exploring-the-possibility-of-using-humanoid-robots

Chang, E. C., D'Zurilla, T. J., & Sanna, L. J. (2004). *Social problem solving: Theory, research, and training*. Washington, DC: American Psychological Assosciation.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, *109*, 162–175.

Chin, C. (2006). Classroom interaction in science: Teacher questioning and feedback to students' responses. *International Journal of Science Education*, *28*(11), 1315–1346.

Church, W. J., Ford, T., Perova, N., & Rogers, C. (2010). Physics with robotics - using LEGO MINDSTORMS in high school education. In *2010 AAAI Spring Symposium Series* (pp. 47–49). Retrieved from https://www.aaai.org/ocs/index.php/SSS/SSS10/paper/viewPaper/1062

Clark, L. A., & Watson, D. (1995). Constructing validity: Basic issues in objective scale development. *Psychological Assessment*, *7*(3), 309–319.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*(1), 37–46.

Cohen, J. (1992). Statistical power analysis. *Current Directions in Psychological Science, 1*(3), 98–101.

Cohen, L., Manion, L., & Morrison, K. (2011). *Research methods in education (7th ed.)*. Abingdon, England: Routledge.

Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: making thinking visible. *American Educator*, *15*(3), 6–11.

Creswell, J. W. (2014). *Research design: Qualitative, quantitative and mixed methods approaches (4th edition)*. Thousand Oaks, CA: Sage Publications.

D'Zurilla, T. J. (1986). *Problem-solving therapy: A social competence approach to clinical intervention*. New York: Springer-Verlag.

D'Zurilla, T. J., & Goldfried, M. R. (1971). Problem solving and behavior modification. *Journal of Abnormal Psychology, 78,* 107–126.

D'Zurilla, T. J., & Maydeu-Olivares, A. (1995). Conceptual and methodological issues in social problem-solving assessment. *Behavior Therapy, 26,* 409–432.

D’Zurilla, T. J., & Nezu, A. M. (1990). Development and preliminary evaluation of the Social Problem-Solving Inventory (SPSI). Psychological Assessment. *A Journal of Consulting and Clinical Psychology*, *2*(2), 156–163.

Duckworth, E. (2005). Critical exploration in the classroom. *The New Educator*, *1*(4), 257–272.

Dybå, T., Kampenes, V. B., & Sjøberg, D. I. (2006). A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, *48*(8), 745–755.

Eguchi, A. (2012). Educational robotics theories and practice: Tips for how to do it Right. In V. I. Barker, B. S., Nugent, G., Grandgenett, N., & Adamchuk (Ed.), *Robots in K-12 education: A new technology for learning* (pp. 1–30). Hershey, PA. https://doi.org/doi:10.4018/978-1-4666-0182-6

Eguchi, A. (2013). Educational robotics for promoting 21st century skills. *Journal of Automation, Mobile Robotics & Intelligent Systems*, *8*(1), 5–11. https://doi.org/10.14313/JAMRIS_1-2014/1

Eguchi, A. (2014). Learning experience through robocupjunior: Promoting engineering and computational thinking skills through robotics competition. In *121st ASEE Annual Conference & Exposition* (pp. 24–41). Retrieved from https://www.asee.org/file_server/papers/attachment/file/0004/4242/LearningExperienceThroughRCJ-ASEE2014_final.pdf

Englehart, J. M. (2009). Teacher-student interaction. In L. Saha & A. G. Dworkin (Eds.), *International handbook of research on teachers and teaching*. Boston,US: Springer. https://doi.org/10.1007/978-0-387-73317-3

Erdfelder, E., Faul, F., & Buchner, A. (1998). GPOWER: A general power analysis program. *Behavior Research Methods, Instruments, & Computers*, *28*(1), 1–11.

Feilzer, M. Y. (2010). Doing mixed methods research pragmatically: Implications for the rediscovery of pragmatism as a research paradigm. *Journal of Mixed Methods Research*, *4*(1), 6–16.

Flake, J. (1990). An exploratory study of Lego. *Journal of Computing in Childhood Education*, *1*(3), 15–22.

Ford, M. J., Dack, G. H., & Prejean, L. (2006). Robotics: Implementing problem based learning in teacher education and field experience. In *Society for Information Technology & Teacher Education International Conference* (pp. 3410–3416). Association for the Advancement of Computing in Education (AACE).

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Educational Research*, *74*(1), 59–109.

Good, T. L., & Brophy, J. E. (1971). Analyzing classroom interaction: A more powerful alternative. *Educational Technology*, *11*(10), 36–41.

Grandgenett, N., Ostler, E., Topp, N., & Goeman, R. (2012). Robotics and problem-based learning in STEM formal educational environments. In B. S. Barker (Ed.), *Robots in K-*

12 education: A new technology for learning (pp. 94–119). Hershey, Pennsylvania: IGI Global. https://doi.org/10.4018/978-1-4666-0182-6

Gravetter, F. J., & Wallnau, L. B. (2014). *Essentials for the behavioral sciences (8th Ed.)*. Boston, the U.S.: Wadsworth Cengage Learning.

Grover, S., & Pea, R. (2010). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Hair, J. (2014). *A primer on partial least squares structural equation modeling (PLS-SEM)*. Thousand Oaks, CA: SAGE Publications.

Hayes, J. R. (1978). *Cognitive psychology thinking and creating*. Homewood, the U.S.: Dorsey Press.

Heppner, P. P., & Petersen, C. H. (1982). The development and implications of a personal problem-solving inventory. *Journal of Counseling Psychology, 29*(1), 66–75.

Heppner, P. P., Witty, T. E., & Dixon, W. A. (2004). Problem-solving appraisal and human adjustment: A review of 20 years of research using the Problem Solving Inventory. *The Counseling Psychologist*, *32*(3), 344–428.

Hussain, S., Lindh, J., & Shukur, G. (2006). The effect of LEGO training on pupils’ school performance in mathematics, problem solving ability and attitude: Swedish data. *Journal of Educational Technology & Society*, *9*(3), 182–194.

International Society for Technology in Education (ISTE). (2011). Operational definition of computational thinking for K-12 education. Retrieved from https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2

Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, *23*(6), 2531–2544.

Ioannou, I., & Angeli, C. (2016). A framework and an instructional design model for the development of students' computational and algorithmic thinking. In *Mediterranean Conference on Information Systems 2016 Proceeding* (pp. 19–25). Retrieved from https://aisel.aisnet.org/mcis2016/19

Järvelä, S. (1999). New models of teacher-student interaction: A critical review. *European Journal of Psychology of Education*, *11*(3), 249–268.

Jeschke, S., Kato, A., & Knipping, L. (2008). The engineers of tomorrow: Teaching robotics to primary school children. In *SEFI Annual Conference 2008*. Retrieved from https://www.semanticscholar.org/paper/The-Engineers-of-Tomorrow-Teaching-Robotics-to-Jeschke-Kato/208ccc2fe3d30ca142aa82f6fc5cc73c6d151728

Johnson, J. (2003). Children, robotics, and education. *Artificial Life and Robotics*, *7*, 16–21. https://doi.org/10.1007/s10015-003-0265-5

Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology Research and Development*, *48*(4), 63–85. https://doi.org/10.1007/BF02300500

Jordan, M. E., & McDaniel Jr, R. R. (2014). Managing uncertainty during collaborative problem solving in elementary school teams: The role of peer influence in robotics engineering activity. *Journal of the Learning Sciences*, *23*(4), 490–536.

Jung, S. E., & Won, E. (2018). Systematic review of research trends in robotics education for young children. *Sustainability*, *10*(905), 1–24. https://doi.org/10.3390/su10040905

Kalu, I., & Ali, A. N. (2004). Classroom interaction patterns, teacher and student characteristics and students' learning outcomes in physics. *The Journal of Classroom Interaction*, *39*(2), 24–31.

Kaushik, V., & Walsh, C. A. (2019). Pragmatism as a research paradigm and its implications for social work research. *Social Sciences*, *8*(9), 255–271.

Kazakoff, E. R., & Bers, M. U. (2008). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Australasian Journal of Educational Technology*, *50*(4), 553–573.

Khanlari, A. (2013). Effects of Robotics on 21st century skills. *European Scientific Journal*, *9*(27), 26–36.

Kim, H. Y. (2013). Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis. *Restorative Dentistry & Endodontics*, *38*(1), 52–54.

King, A. (2002). Structuring peer interaction to promote high-level cognitive processing. *Theory into Practice*, *41*(1), 33–39.

Kucuk, S., & Sisman, B. (2017). Computers & education behavioral patterns of elementary students and teachers in one-to-one robotics instruction. *Computers & Education*, *111*, 31–43. https://doi.org/10.1016/j.compedu.2017.04.002

Larkins, D. B., Moore, J. C., Rubbo, L. J., & Covington, L. R. (2013). Application of the cognitive apprenticeship framework to a middle school robotics camp. In *Proceeding of the 44th ACM technical symposium on computer science education* (pp. 89–94). Retrieved from https://dl.acm.org/doi/10.1145/2445196.2445226

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., … Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37.

Lee, K. T., Sullivan, A., & Bers, M. U. (2013). Collaboration by design: Using robotics to foster social interaction in kindergarten. *Computers in the Schools*, *30*(3), 271–281. https://doi.org/10.1080/07380569.2013.805676

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, *25*(6), 860–876. https://doi.org/10.1007/s10956-016-9628-2

Levy, S. T., & Mioduser, D. (2010). Approaching complexity through planful play: Kindergarten children's strategies in constructing an autonomous robot's behavior. *Journal of Computers for Mathematical Learning*, *15*(1), 21–43.

Li, Y., Yang, B., & Chen, Z. (2019). 小學生計算思維培養的過程和策略研究—基於對武漢市從事機器人教育的 26 位教師的深度訪談[Learning powered by technology from perspective of cost-effectiveness analysis: Based on survey of minors' new media use in 2018] (in Chinese). *E-Education Research*, *40*(12), 115–121.

Lin, C. (2017). 构建中国化的学生发展核心素养 [To construct sinicized core competencies and values for student development] (in Chinese). *Journal of Beijing Normal University（Social Sciences）*, (1), 66–73.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Thousand Oaks, California: Sage.

Liu, E. Z. F., Lin, C. H., Liou, P. Y., Feng, H. C., & Hou, H. T. (2013). An analysis of teacher-student interaction patterns in a robotics course for kindergarten children: A pilot study. *Turkish Online Journal of Educational Technology*, *12*(1), 9–18.

Maloney, J., Peppler, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with Scratch. In *Proceedings of SIGCSE'08*. New York: ACM Press.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, *10*(4), 1–15. https://doi.org/10.1145/1868358.1868363.http

Markee, N. (2015). *The handbook of classroom discourse and interaction (Vol. 115)*. Hoboken, United States: John Wiley & Sons.

Martinez, M. E. (1998). What is problem solving? *The Phi Delta Kappan*, *79*(8), 605–609.

Mattheos, N. (2004). *Information technology and interaction in learning (Doctoral thesis)*. Malmö University, Malmö, Sweden.

Maydeu-olivares, A., & D’Zurilla, T. J. (1997). The factor structure of the Problem Solving Inventory. *European Journal of Psychological Assessment*, *13*(3), 206–215.

Merriam, S. B. (1998). *Qualitative research and case study applications in education*. San Francisco: Jossey-Bass.

Messer, D. J., Joiner, R., Loveridge, N., Light, P., & Littleton, K. (1993). Influences on the effectiveness of peer interaction: Children’s level of cognitive development and the relative ability of partners. *Social Development*, *2*(3), 279–294.

Ministry of Education. (2017a). *普通高中信息技术课程标准 [New curriculum standards for senior high school: Information technology] (in Chinese)*. Beijing, China: The People’s Education Publishing Company.

Ministry of Education. (2017b). *普通高中通用技术课程标准 [New curriculum standards for senior high school: General technology](in Chinese)*. Beijing, China: The People’s Education Publishing Company.

Mitnik, R., Recabarren, M., Nussbaum, M., & Soto, A. (2009). Collaborative robotic instruction: A graph teaching experience. *Computers and Education*, *53*(2), 330–342. https://doi.org/10.1016/j.compedu.2009.02.010

Moore, G. M. (1989). Three types of interaction. *The American Journal of Distance Education*, *3*(2), 1–7.

Morgan, D. L. (2014). Pragmatism as a paradigm for social research. *Qualitative Inquiry*, *20*(8), 1045–1053.

Mosley, P., Ardito, G., & Scollins, L. (2016). Robotic cooperative learning promotes student STEM interest. *American Journal of Engineering Education*, *7*(2), 117–128. Retrieved from https://eric.ed.gov/?id=EJ1131333

Mubin, O., Stevens, C. J., Shahid, S., Mahmud, A. Al, & Dong, J.-J. (2013). A review of the applicability of robots in education. *Technology for Education and Learning*, *1*(1). https://doi.org/10.2316/Journal.209.2013.1.209-0015

Nelson, C. A. (2012). Generating transferable skills in STEM through educational robotics. In Bradley S. Barker, N. Gwen, G. Neal, & I. A. Viacheslav (Eds.), *Robots in K-12 education: A new technology for learning* (pp. 54–65). Hershey, Pennsylvania: IGI Global.

Netemeyer, R. G., Bearden, W. O., & Sharma, S. (2003). *Scaling procedures*. Newbury Park, California: SAGE Publishing. https://doi.org/http://dx.doi.org.ezproxy.eduhk.hk/10.4135/9781412985772

Norton, S. J., McRobbie, C. J., & Ginns, I. S. (2007). Problem solving in a middle school robotics design classroom. *Research in Science Education*, *37*(3), 261–277.

Nugent, G., Barker, B., Grandgenett, N., & Adamchuk, V. I. (2010). Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes. *Journal of*

Research on Technology in Education, *42*(4), 391–408. https://doi.org/10.1080/15391523.2010.10782557

OECD. (2013). *PISA 2012 problem-solving framework*. *Programme for International Student Assessment (PISA)*. Paris: PISA, OECD Publishing. https://doi.org/10.1787/19963777

Office of the Press Secretary. (2011). *President Obama launches advanced manufacturing partnership*. Retrieved from https://obamawhitehouse.archives.gov/the-press-office/2011/06/24/president-obama-launches-advanced-manufacturing-partnership

Olabe, J. C., Olabe, M. A., Basogain, X., & Castaño, C. (2010). Programming and robotics with Scratch in primary education. In A. Méndez-Vilas (Ed.), *Education in a technological world: communicating current and emerging research and technological efforts* (pp. 356–363). Badajoz, Spain: FORMATEX 2011.

Ömeroğlu, E., Büyüköztürk, Ş., Aydoğan, Y., & Özyürek, A. (2009). Determining the views of preschool and primary school teachers over the support of problem solving skills at children. *Procedia-Social and Behavioral Sciences*, *1*(1), 1969–1974.

Onwuegbuzie, A. J., & Johnson, R. B. (2006). The validity issue in mixed research. *Research in the Schools*, *13*(1), 48–63.

Papert, S. (2001). Change and resistance to change in education: Taking a deeper look at why school hasn't changed. Retrieved from http://dailypapert.com/wp-content/uploads/2018/06/Papert-Lisbon-Paper.1-1.pdf

Petre, M., & Price, B. (2004). Using robotics to motivate ‘Back Door’ learning. *Education and Information Technologies*, *9*(2), 147–158. https://doi.org/10.1023/B:EAIT.0000027927.78380.60

Pianta, R. C., Hamre, B. K., & Allen, J. P. (2012). Teacher-student relationships and engagement: Conceptualizing, measuring, and improving the capacity of classroom interactions. In S. L. Christenson, A. L. Reschly, & C. Wylie (Eds.), *Handbook of research on student engagement* (pp. 365–386). Boston, MA: Springer.

Polanyi, M. (2009). *The tacit dimension (7th edition)*. Chicago: University of Chicago press.

Polya, G. (2004). *How to solve it: A new aspect of mathematical method*. Princeton, New Jersey: Princeton University Press.

Price, A., Rimington, R., Chew, M. T., & Demidenko, S. (2010). Project-based learning in robotics and electronics in undergraduate engineering program setting. In *2010 Fifth IEEE International Symposium on Electronic Design* (pp. 188–193).

Prince, S. A., Adamo, K. B., Hamel, M. E., Hardt, J., Gorber, S. C., & Tremblay, M. (2008). A comparison of direct versus self-report measures for assessing physical activity in adults: A systematic review. *International Journal of Behavioral Nutrition and Physical Activity*, *5*(1), 56–79.

Rardin, D. R., & Moan, C. E. (1971). Peer interaction and cognitive development. *Child Development*, *42*, 1685–1699.

RAS-SIG Steering Group. (2014). *RAS 2020 Robotics and Autonomous Systems*. Retrieved from https://connect.innovateuk.org/documents/2903012/16074728/RAS UK Strategy?version=1.0

Ricca, B., Lulis, E., Bade, D., & Clark, J. S. (2006). *LEGO mindstorms and the growth of critical thinking*. Retrieved from http://www.roboticscamp.ca/documents/papers/critical2006.pdf

Rogers, C., & Portsmore, M. (2004). Bringing engineering to elementary school. *Journal of STEM Education*, *5*(3&4), 17–28.

Schneider, I., Mädler, M., & Lang, J. (2019). Comparability of self-ratings and observer ratings in cccupational psychosocial risk assessments: Is there agreement? *BioMed Research International*, *2019*, 1–10.

Shimabukuro, G. (1989). A class act: Junior high students-LEGO and Logo. *The Computing Teacher*, *16*(5), 37–39.

Shin, N., & Kim, S. (2007). Learning about, from, and with robots: Students' perspectives. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication* (pp. 1040–1045). https://doi.org/10.1109/ROMAN.2007.4415235

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*(1), 3–20.

Sullivan, F. R. (2008). Robotics and science literacy: Thinking skills, science process skills and systems understanding. *Journal of Research in Science Teaching*, *45*(3), 373–394.

Sullivan, G. M., & Feinn, R. (2012). Using effect size—or why the P value is not enough. *Journal of Graduate Medical Education*, *4*(3), 279–282.

The Headquarters for Japan’s Economic Revitalization. (2015). *New robot strategy*. Retrieved from https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/robot_honbun_150210EN.pdf

The State Council Information Office. (2016). *机器人产业发展规划(2016-2020年)[Robot Industry Development Plan (2016-2020)]*. Retrieved from http://www.scio.gov.cn/xwfbh/xwbfbh/wqfbh/33978/34888/xgzc34894/Document/1484894/1484894.htm

The State Council of China. (2015). *中国制造2025 [Made in China 2025]*. Retrieved from http://www.gov.cn/zhengce/content/2015-05/19/content_9784.htm

The State Council of China. (2017). *新一代人工智能发展规划 [Next-generation Artificial Intelligence Development Plan]*. Retrieved from http://www.gov.cn/zhengce/content/2017-07/20/content_5211996.htm

Tian, L., Heppner, P. P., & Hou, Z. (2014). Problem solving appraisal and its relationship to career decision-making difficulties and psychological adjustment in China. *International Perspectives in Psychology: Research, Practice, Consultation*, *3*(1), 19.

Toh, E. L. P., Causo, A., Tzuo, P. W., Chen, I. M., & Yeo, S. H. (2016). A review on the use of robots in education and young children. *Educational Technology & Society, 19*(2), 148–163. https://doi.org/10.2307/jeductechsoci.19.2.148

Tsei, C. F. (2009). *Lego Mindstorms提升國小學童問題解決能力與科學態度之研究 [A study of applying Lego Mindstorms learning activities to promote problem solving ability and scientific attitude for elementary school students]（Master's thesis）.* National Taitung University, Taitung County, Taiwan.

Varnado, T. E. (2005). *The effects of a technological problem solving activity on first lego league participants' problem solving style and performance (Doctoral dissertation)*. Virginia Tech, Blacksburg, the U.S.

Wagner, S. P. (1998). Robotics and children: Science achievement and problem solving. *Journal of Computing in Childhood Education*, *9*(2), 149–176.

Wang, E., LaCombe, J., & Vollstedt, A. M. (2008). *Teaching structured programming using Lego programmable bricks* (Vol. 18).

Wang, M. C., Haertel, G. D., & Walberg, H. J. (1990). What influences learning? A content analysis of review literature. *The Journal of Educational Research*, *84*(1), 30–43.

Weinberg, J. B., Pettibone, J. C., Thomas, S. L., Stephen, M. L., & Stein, C. (2007). *The impact of robot projects on girls' attitudes toward science and engineering. Workshop on research in robots for education* (Vol. 3).

Williams, D. C., Ma, Y., Prejean, L., Ford, M. J., & Lai, G. (2007). Acquisition of physics content knowledge and scientific inquiry skills in a robotics summer camp. *Journal of Research on Technology in Education*, *40*(2), 201–216.

Winer, L. R. (1989). Children in an educational robotics environment. In *MECC 89 conference*. Minneapolis, MN.

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

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725.

Wing, J. M. (2010). *Computational Thinking: What and Why?* Retrieved from https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Wong, W. H., Chew, E., & Wong, S. M. (2016). The review of educational robotics research and the need for real-world interaction analysis. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (pp. 1–6). Retrieved from https://ieeexplore.ieee.org/document/7838707/

Xia, L., & Zhong, B. (2018). A systematic review on teaching and learning robotics content knowledge in K-12. *Computers and Education*, *127*(122), 267–282. https://doi.org/10.1016/j.compedu.2018.09.007

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. In *Proceedings 42nd ACM Technical*

Symposium on Computer Science Education (pp. 465–470). New York: ACM Press. https://doi.org/10.1145/1953163.1953297

Yang, J., & Qiao, F. (2017). *中小学机器人教育调研报告 [K-12 Robotics Education research report](in Chinese)*. Retrieved from http://kp.cie-info.org.cn/h-nd-159.html

Yuen, T. T., Boecking, M., Tiger, E. P., Gomez, A., Guillen, A., Arreguin, A., & Stone, J. (2014). Group tasks, activities, dynamics, and interactions in collaborative robotics projects with elementary and middle school children, *15*(1), 39–46.

Zainal, N. F. A., Abdullah, S. N. H. S., & Prabuwono, A. S. (2012). Adapting robot soccer game in student self-centered learning. *Procedia - Social and Behavioral Sciences*, *59*, 130–137. https://doi.org/10.1016/j.sbspro.2012.09.256

Zhang, G., & Zhang, J. (2008). The issue of robot education in China’s basic education and its strategies. In *2008 IEEE International Conference on Robotics on Automation and Mechatronics* (pp. 702–705). https://doi.org/10.1109/RAMECH.2008.4681440

Zhao, X. H. (2018). *基于项目式学习的小学机器人课程设计研究-以＊＊小学的ＶＥＸＩＱ机器人教育课程为例 [A study of project-based robotics curriculum design in primary school: A case study of one primary school’s VEXIQ robotics learning curriculum](In Chinese)*. Shaanxi Normal University, Xi’an, China.

Zhou, J., & Ye, J. (2019). 推进机器人教育:背景、定位及可能路径 [Advancing Robot Education:Background, Positioning and Possible Paths]. *Digital Education*, *5*(3), 15–20.

Zohrabi, M. (2013). Mixed method research: instruments, validity, reliability and reporting findings. *Theory & Practice in Language Studies*, *3*(2), 254–262.

Appendix A: Summary of Relevant RE Studies

RE studies	Participants	Methods	About classroom interactions	About learning outcomes (i.e., PS and CT)
Kucuk and Sisman (2017): *Behavioural patterns of elementary students and teachers in one-to-one robotics instruction*	18 primary school students (age 8-11) and 18 preservice teachers.	Quantitative content analysis and lag sequential analysis based on videotaped data.	Patterns of one-to-one S-T interactions	Finding and solving problems was coded as one of students' behaviours, but the frequency of this behaviour was ranked on the last place among four students' behaviours
Liu et al. (2013): *An analysis of teacher-student interaction patterns in a robotics course for kindergarten children: A pilot study*	One kindergarten child (age 5) and one kindergartener.	Quantitative content analysis and lag sequential analysis based on videotaped data.	Patterns of one-to-one S-T interactions	Finding and solving problems was coded as one of students' behaviours and it was ranked as the third place based on its frequencies.
Larkins et al. (2013):	Middle school students in a summer camp (no	Quantitative method (pre-/post-test) and	S-T interactions demonstrated by	Students learned PS and CT.

Application of the cognitive apprenticeship framework to a middle school robotics camp	specific number was provided)	qualitatively method (students' works in engineering notebooks)	cognitive apprenticeship teaching model.	
Mosley et al. (2016): *Robotic cooperative learning promotes student STEM interest*	94 middle school students	Quantitative method (pre-/post-test)	S-T interactions; Comparing two pedagogic methods: cooperative learning and problem-based learning.	Not about PS or CT, but about student' critical thinking.
Yuen et al. (2014): *Group tasks, activities, dynamics, and interactions in collaborative robotics projects with elementary and middle school children*	70 primary and middle school students in 18 groups from a summer camp	Quantitative and qualitative methods analysing classroom observation	S-S interactions within group and with others; S-T interactions; Students' behaviours of handing on the robot (not an actual S-R interaction)	N/A
Lee et al. (2013): *Collaboration by design: Using robotics to foster Social interaction in kindergarten*	19 kindergarten children from a summer camp	Self-report and classroom observation in quantitative way	Compared frequencies of S-S interactions in different kinds of S-T interactions: unstructured classroom and structured classroom	N/A
Jordan & McDaniel Jr (2014): *Managing uncertainty during collaborative problem solving in*	24 primary school students	Qualitative method (classroom observation)	S-S interactions about how students responded to their group members'	When students received supportive peer responses, they would

primary school teams: The role of peer influence in robotics engineering activity			attempt to manage uncertainty	participate in collaborative idea and problem solving
Levy & Mioduser (2010): *Approaching complexity through planful play: Kindergarten children's strategies in constructing an autonomous robot's behaviour*	Six children on age five	Qualitative methods (classroom observation and interviews	Students' participatory investigations, which is a kind of bodily S-R interaction was found	N/A
Wagner (1998): *Robotics and children: Science achievement and problem solving*	364 primary school students	Quantitative method (pre-/post-test)	N/A	Students' problem-solving skill in three modes of learning environments were compared.
Norton et al. (2007): *Problem solving in a middle school robotics design classroom*	19 middle students and two teachers	Interpretive study	S-T interactions: "Teachers' goals and beliefs mediated the formation of different learning communities" (p. 265)	It asserted that "more holistic problem solving approaches were associated with students' systemic use of flow charting" (p. 268)
Castledine & Chalmers (2011): *LEGO Robotics: An authentic problem solving tool?*	23 primary school students	Descriptive qualitative case study	N/A	Demonstrated students' problem-solving strategies in RE.

Atmatzidou et al. (2018): *How does the degree of guidance support students' metacognitive and problem solving skills in educational robotics?*	52 primary and middle school students	Quantitative method (pre-/post-test) and qualitative methods (observation, think-aloud protocol, semi-structured interviews)	S-T interactions: compared "strong" and "minimal" teacher's guidance	Students' problem-solving skills in two kinds of teacher's guidance were compared.
Barak and Zadok (2009): *Robotics projects and learning concepts in science, technology and problem solving*	In the first year of the study, 80 students participated; In the second year, 76 students; In the third year, 192 including students participating in the second year's study.	Qualitative methods (class journals, conversation documents, computer file records, photographs, videotaped selective classrooms, and discussions)	N/A	Student usually generate inventive solutions when solving robotics problems.
Barak & Assal (2018): *Robotics and STEM learning: Students' achievements in assignments according to the P3 Task Taxonomy - practice, problem solving, and projects*	32 junior high school students	Qualitative methods (interviews and observation) and quantitative	N/A	In the *P3 Task Taxonomy* proposed by this paper, problem-solving was the second level of assignment.

		methods (exams and questionnaires)		Results indicated that around 50% students can complete tasks at problem-solving level by independent works
Zhao (2018): *A study of project-based robotics curriculum design in primary school: A case study of one primary school's VEXIQ robotics learning curriculum* In Chinese: 基於項目式學習的小學機器人課程設計研究-以＊＊小學的VEXIQ 機器人教育課程為例	21 primary school students	Qualitative method (observation notes) and quantitative method (pre-/post-tests) were used	N/A	Students’ problem-solving skill was measured, and the pre-post result indicated a positive shift.
Tsei (2009): *A study of applying Lego Mindstorms learning activities to promote problem solving ability and scientific attitude for elementary school students [Lego Mindstorms* In Chinese: Lego Mindstorms 提升國小學童問題解決能力與科學態度之研究	42 primary school students	Both qualitative method (feedback questionnaire) and quantitative method (pre-/post-tests) were used	N/A	Problem-solving skill measure of students learning robotics (experimental group) was higher than of the control group.

Hussain et al. (2006): *The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data*	322 middle school students	Both qualitative methods (observation, interview and inquiry) and quantitative methods (tests, inquiries) were used.	N/A	PS skill was measured; however, no significant results were found between controlled and trained groups.
Lee et al. (2011): *Computational thinking for youth in practice*	N/A	Descriptive study based on researchers' experiences	N/A	It discussed how CT looks like in RE practice.
Bers (2010) : *The TangibleK robotics program: Applied computational thinking for young children*	Teachers and children from kindergarten to Grade-2; no exact number was offered.	Descriptive study for introducing a robotics program called TangibleK	N/A	It provided a framework to apply computational thinking in RE and tools for evaluating students' CT.
Bers et al. (2014): *Computational thinking and tinkering: Exploration of an early childhood robotics curriculum*	53 kindergarten children	Design-based research; quantitative data collection (pre-/post-test).	N/A	Students' CT was measured.

Eguchi (2014): *Learning experience through robocupjunior: Promoting engineering and computational thinking skills through robotics competition*	168 participants (age 10-19) of RoboCupJunior (RCJ).	Survey study	N/A	Most of participants (above 60%) agreed with that they learned CT when preparing and participating in RCJ
Ioannou & Angeli (2016): *A framework and an instructional design model for the development of students' computational and algorithmic thinking*	248 middle school students	Quantitative method (pre-/post-test)	S-T interactions; Teaching intervention was based on Technological Pedagogical Content Knowledge (TPCK) and Technology Mapping (TM) models.	It measured and compared students' CT in two different groups: robotics learning guided by TPCK and TM, and 3D interactive programming.
Leonard et al. (2016): *Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills*	126 middle school students were recruited; 101 out of 126 students attended classroom observation; but only 76 complete both pre- and post- tests.	Quantitative method (pre-/post-tests) and qualitative methods (field notes, screenshots, computer files, and photographs of student work and artifacts)	N/A	Students' CT skill was observed and measured by a CT rubric.

Atmatzidou & Demetriadis (2016): *Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences*	164 students from junior high and high vocational schools	Quantitative methods (pre-/post-questionnaire, think-aloud protocol, and opinion questionnaire) and qualitative methods (interview and observation)	N/A	CT skill was measured.
Chen et al. (2017): *Assessing elementary students' computational thinking in everyday reasoning and robotics programming*	121 primary school students participated in the pre-exam, but only 37 of them participated in the post-exam.	Quantitative methods (Pre-/post-exams and scoring for the open-ended questions)	N/A	CT skill was measured.
Li, Yang, & Chen (2019): *Learning powered by technology from perspective of cost-effectiveness analysis:Based on survey of minors' new media use in 2018* In Chinese: 小學生計算思維培養的過程和策略研究—基於對武漢市從事機器人教育的 26 位教師的深度訪談	26 primary school teachers	Qualitative method (semi-structure interview)	S-T interactions; Teaching process was discussed in four stages and eight links.	It discussed how students' CT was cultivated.

Appendix B: Shorter Problem-Solving Inventory (SPSI, Maydeu-olivares & D’Zurilla, 1997, p. 215)

Problem-Solving Self-Efficacy (PSSE)

1. When my first efforts to solve a problem fail, I become uneasy about my ability to handle the situation.
2. I have the ability to solve most problems even though initially no solution is immediately apparent.
3. Many of the problems I face are too complex for me to solve.
4. When I make plans to solve a problem, I am almost certain that I can make them work.
5. Given enough time and effort, I believe I can solve most problems that confront me.
6. When faced with a novel situation, I have confidence that I can handle problems that may arise.
7. I trust my ability to solve new and difficult problems.

Problem-Solving Skills (PSS)

8. When a solution to a problem has failed, I do not examine why it didn’t work.
9. After following a course of action to solve a problem, I compare the actual outcome with the one I had anticipated.
10. When I have a problem, I think of as many possible ways to handle it as I can until I can’t come up with any more ideas.
11. When considering solutions to a problem, I do not take the time to assess the potential success of each alternative.
12. When confronted with a problem, I stop and think about it before deciding on a next step.
13. When making a decision, I compare alternatives and weigh the consequences of one against the other.
14. I try to predict the result of a particular course of action.
15. When thinking of ways to handle a problem, I seldom combine ideas from various alternatives to arrive at a workable solution.
16. When confronted with a problem, I usually first survey the situation to determine the relevant information.

Appendix C: The Chinese Version of SPSI Used in This Research (CSPSI, adapted from Maydeu-olivares and D’Zurilla [1997])

	非常不同意	不同意	比较不同意	比较同意	同意	非常同意
1. 当[我/这个学生]为解决一个问题所作的第一次努力失败时，[我/这个学生]会对自己处理事情的能力感到不安。	1	2	3	4	5	6
2. 即使一开始未能立刻找到解决问题的方法，[我/这个学生]仍相信[我/这个学生]有能力去解决大部分的问题。	1	2	3	4	5	6
3. [我/这个学生]遇到的许多问题都太复杂，[我/这个学生]没办法解决。	1	2	3	4	5	6
4. 当[我/这个学生]做计划解决一个问题时，[我/这个学生]几乎有信心[我/这个学生]可以使[我/这个学生]的计划行得通。	1	2	3	4	5	6
5. 只要有充分的时间与努力，[我/这个学生]相信[我/这个学生]可以解决大多数[我/这个学生]所面对的问题。	1	2	3	4	5	6
6. 当面对一个新的情况时，[我/这个学生]有信心自己有能力处理可能会产生的问题。	1	2	3	4	5	6
7. [我/这个学生]相信自己具有解决新问题与困难问题的能力。	1	2	3	4	5	6
8. 在[我/这个学生]采取一些做法去解决问题之后，[我/这个学生]会比较实际的结果与预期的结果。	1	2	3	4	5	6
9. 当[我/这个学生]有一个问题时，[我/这个学生]会尽量想出所有可能解决问题的方法去处理它，直到[我/这个学生]无法再想出其他的点子为止。	1	2	3	4	5	6
10. 当面对一个问题时，[我/这个学生]会先停下来想想之后，才决定下一个步骤。	1	2	3	4	5	6
11. 当作一个决定时，[我/这个学生]会比较每种方法并权衡轻重。	1	2	3	4	5	6
12. [我/这个学生]会尝试去预测采取某种行动后的结果。	1	2	3	4	5	6
13. 当[我/这个学生]思考如何处理一个问题的时候，[我/这个学生]很少会将不同的备选方案结合在一起去形成一个可行的方案。	1	2	3	4	5	6
14. 当面对一个问题时，[我/这个学生]通常会先调查清楚情况去确定相关的信息。	1	2	3	4	5	6

Appendix D: Computational Thinking Rubric (CTR, adapted from ISTE (2014) and Leonard et al. (2016))

	初步显现（1 分） **Emerging (1)**	**中等水平（2 分）** **Moderate (2)**	**大量显现（3 分）** **Substantive (3)**
问题阐述能力： **学生能将问题阐述为一种可以用电脑或其它工具去解决的形式。这个过程可能需要运用到问题分解。** **Formulating problems:** **Formulating problems in a way that enables us to use a computer and other tools to help solve them. It may involve problem decomposition.**	阐述的问题过于宽泛以至于很难通过编程来解决，或者阐述的问题与本课任务目标无关。 例如：“如何制作机器人？” “猪会飞吗” Problems that too general to be solved by computer programming OR irrelated to task goal. e.g.: “How to make a robot?” “Can pigs fly?”	阐述的问题基本与本课任务目标相关，但是太过宽泛以至于很难通过编程来解决。 例如：“如何制作一个足球机器人？” Problems that basically related to task goal BUT too general to be solved by computer programming. e.g.: “How to make a football robot?”	阐述的问题与本课任务目标相关且足够具体，使得这个问题有可能通过编程来解决。 例如：“如何命令机器人去探测是否触到了墙？” Problems that related to task goal AND specific enough so could be possibly solved by computer programming. e.g.: “How to order the robot to detect whether it touches the wall?”
抽象能力： **学生能通过抽象化的方式，例如模型和模拟，去表现数据。** **Abstraction:** **Representing data through abstractions such as models and simulations**	学生没有通过抽象的方法来表现问题。 No evidence of interpreting problems in an abstractive way.	学生通过抽象的方法来表现问题，但比较有限。 例如：“如果机器人触到了墙，那么它应该回头。” Limited evidences of interpreting problems in an abstractive way. e.g.: “If the robot touches the wall, then it should return back.”	学生可以适当地通过抽象的方法来表现问题。 例如：“如果机器人从触碰传感器接收到信号 1，那么马达转数的数值应该变为负值。” Can properly interpret problems in an abstractive way. e.g.: “If the robot receives an ‘1’ sign from touch sensor, then the parameters of motor speed should be negative.”
逻辑思维能力：	学生的程序表达没有任何逻辑。	学生的程序表达有逻辑，并且有一些复杂度。	学生的程序表达有逻辑，并且有比较多的复杂

学生能够有逻辑的去组织和分析数据。 **Logical thinking:** **Logically organizing and analysing data**	例如：机器人完全失控。 Statements do not follow logical path e.g.: The robot is out of control.	例如：机器人能基本完成本课任务。 Statements follow logical path with some complexity. e.g.: The robot basically completes the given task.	度。 例如：机器人能完成挑战性的任务。 Statements follow logical path with more complexity. e.g.: The robot can complete challenging task(s).
使用算法的能力： **学生能通过算法思考来实现问题的解决方案。** **Using algorithm:** **Automating solutions through algorithmic thinking (a series of ordered steps)**	没有证据显示学生合理运用算法。 例如：不能合理运用任何编程逻辑表达式（例如，条件逻辑，重复逻辑，或者平行逻辑） No evidence of using proper algorithm. e.g.: Failed to use proper programming logics (i.e., conditional logic, iterative logic, or parallel logic).	有部分证据显示学生合理运用算法。 例如：能够合理运用一种编程逻辑表达式。 Some evidence of using proper algorithm. e.g.: Properly use only one programming logic.	较多证据显示学生合理运用算法。 例如：能够运用两种或以上编程逻辑表达式。 More evidence of using proper algorithm. e.g. Properly use two or more programming logics.
分析和实施解决方案的能力： **学生能够以整合最有效率、最有用的步骤与资源为目标，去确认、分析、实施可能的解决方案。** **Analysing and implementing solutions:** **Identifying, analysing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources**	没有证据显示学生有能力与调试程序。 No evidence of the ability to debug the program.	部分证据显示学生进行了调试。 Some evidence of debugging.	很强的证据显示学生进行了调试。 Strong evidence of debugging
归纳以及问题转换的能力： **学生能够归纳并转换问题解决过程去解决多种**	学生经常被同类的问题困住。 Always get stuck by similar problems.	学生会被新问题困住但是能将问题解决过程运用到同类问题中。	学生能够将问题解决过程运用到多种多样的问题中。

多样的问题。 **Generalizing and problem transfer: Generalizing and transferring this problem-solving process to a wide variety of problems**		Stuck by novel problems but can transfer problem-solving process to similar problems.	Can transfer problem-solving process to a wide variety of problems.

Appendix E: Coding Scheme

1 Student-Teacher interactions

1.1 Teachers' behaviors

1.1.1 Modeling

1.1.2 Coaching

1.1.3 Scaffolding

1.1.4 Articulation

1.1.5 Reflection

1.1.6 Exploration

1.2 Student's behaviors

1.2.1 Active engagement

1.2.2 Inactive engagement

2 Student-student interaction

2.1 Cooperative interactions

2.1.1 Discussion

2.1.2 Observational learning

2.2 Off-task interactions

3 Student-robot interaction

3.1 Building

3.2 Programming

3.3 Observing

3.4 Participatory investigation

4 The features of problem-solving

4.1 Kinds of problems

4.2 Problem-solving process

4.3 How students generate solutions

5 Problem-solving competencies

5.1 Problem-solving self-efficacy

5.2 Problem-solving skills

5.2.1 Problem definition and formulation

5.2.2 Generation of alternative solutions

5.2.3 Decision making

5.2.4 Solution implementation and verification

6 The features of computational thinking

6.1 Kinds of problems

6.2 Computational thinking process

6.3 Relationship between PS and CT

7 Computational thinking skills

7.1 Formulating problems

7.2 Abstraction

7.3 Logical thinking

7.4 Using algorithm

7.5 Analyzing and implementing solutions

7.6 Generalizing and problem transfer

Appendix F: Interview Protocol (for the Teacher)

Part one: Profile information

1. Could you please introduce yourself? (e.g.: age, certificates awarded, experiences, reason of joining in this summer camp, etc.)

Part two: About the student-teacher interactions

1. How do you describe your teaching behaviors/process in the summer camp?
2. How did you implement cognitive apprenticeship teaching methods?
3. What did you do when you were modeling? How was student’s engagement when you were modeling? Do you have any examples of active or inactive engagement?
4. What did you do when you were coaching students? How was student’s engagement when you were coaching? Do you have any examples of active or inactive engagement?
5. What did you do when you were scaffolding students? How was student’s engagement when you were scaffolding? Do you have any examples of active or inactive engagement?
6. What did you do when you attempt to make students to articulate their thinking? How was student’s engagement correspondingly? Do you have any examples of active or inactive engagement?
7. What did you do when you were guiding student’s reflection? How was student’s engagement at that time? Do you have any examples of active or inactive engagement?
8. What did you do when you were supporting student’s exploration? How was student’s engagement? Do you have any examples of active or inactive engagement?
9. Do you think S-T interactions mentioned above were effective for the development of student’s PS and CT? Why? For example?
10. Can you talk about some critical moments between you and your students? What were your and students’ words/behaviors/movements at that moment?
11. Do you have any other experiences or ideas you are willing to share with me about interactions between you and your students?

Part three: About the peer interactions

1. Have you organized group/pair activities in the summer camp? Why?
2. Can you classify the kinds of student’s peer interactions? For example? Which kinds of peer interactions do you think were effective for the development of student’s PS and CT? Why? For example?
3. Have you ever noticed student’s discussions? Do you have any examples? Do you think they learned from such discussions? Why? For example?
4. Have you ever noticed student’s peer observation? Do you have any examples? Do you think they learned from such observation? Why? For example?
5. Have you ever noticed student’s off-task interactions? Do you have any examples? Do you think they learned from such off-task interactions? Why? For example?
6. What was your role in student’s peer interactions?
7. Do you have any other experiences or ideas you are willing to share with me about interactions among students?

Part four: About the student-robot interaction

1. Do you think there is an interaction between student and robot? Why? For example?
2. Can you classify the kinds of S-R interactions? For example? Which ones do you think were effective for the development of student's PS and CT? Why? For example?
3. Have you ever noticed any S-R interactions when students were building a robot? For example? Do you think they learned from such interactions? Why? For example?
4. Have you ever noticed any S-R interactions when students were programming for a robot? For example? Do you think they learned from such interactions? Why? For example?
5. Have you ever noticed any S-R interactions when students were observing a robot's behaviors? For example? Do you think they learned from such interactions? Why? For example?
6. Have you ever noticed students were bodily participating in S-R interactions? For example? Do you think they learned from such interactions? Why? For example?
7. What were the roles of you and peers in S-R interactions? For example?
8. Do you have any other experiences or ideas you are willing to share with me about student-robot interaction?

Part five: About the problem-solving competencies

1. How do you see the problem-solving self-efficacy and skills? To what extent you understand these concepts?
2. Could you describe the features of students' problem-solving according to your observation?
3. Do you think students' problem-solving self-efficacy was changed? Why? For example (Any evidences)?
4. Do you think students' problem-solving skills (Problem definition and formulation, Generation of alternative solutions, Decision making, Solution implementation and verification) were changed? Why? For example (Any evidences)?
5. Do you think student's change in problem-solving self-efficacy/skills was influenced by S-T/S-S/S-R interactions? Why? For example?

Part six: About the computational thinking skills

1. How do you see the computational thinking skills? To what extent you understand this concept?
2. Have you ever noticed any features of student's computational thinking? For example?
3. Do you think students' computational thinking skills (Formulating problems, Abstraction, Logical thinking, Using algorithms, Analysing and implementing solutions, Generalizing and problem transfer) were changed? Why? For example (Any evidences)?
4. Do you think student's change in problem-solving self-efficacy/skills was influenced by S-T/S-S/S-R interactions? Why? For example?

Appendix G: Interview Protocol (for Students)

Part one: profile information

1. Could you please introduce yourself? (age, grade, experiences, the reason of joining in the summer camp, etc.)

Part two: Student-teacher interactions

1. Do you think there were any S-T interactions in the summer camp? How often? When and how? For example?
2. What kinds of S-T interactions do you think were helpful for your learning? For example?
3. Do you think you have actively engaged in S-T interaction? Why? For example?
4. Did you learn from S-T interactions? In what aspects? Why? For example?
5. Do you have any critical moments with your teachers? For example?

Part three: Student-student interactions

1. Do you think there were any S-S interactions in the summer camp? How often? When and how? For example?
2. What kinds of S-S interactions do you think were helpful for your learning? For example?
3. Did you learn from S-S interactions? In what aspects? Why? For example?
4. Do you have any critical moments with your classmates? For example?

Part four: Student-robot interaction

1. Do you think there were any S-R interactions in the summer camp? How often? When and how? For example?
2. What kinds of S-R interactions do you think were helpful for your learning? For example?
3. Did you learn from S-R interactions? In what aspects? Why? For example?
4. Do you have any critical moments with your robots? For example?

Part five: Problem-solving competencies

1. Do you know what is problem-solving self-efficacy/skills? Can you describe these two terms based on your own experiences?
2. Can you classify the kinds of problems you encountered in the summer camp? For example?
3. How did you solve those problems? Could you describe the process when you were solving a problem?
4. What were your attitudes or feelings when you met a problem? Why? For example?
5. Did you think you have confidence to solve problems? Why? For example? Do you think you became more and more confident when solving problem? Why? For example?
6. How did you define/formulate a problem? For example? Do you think your ability of defining/formulating problems was changed? Why? For example?
7. How did you generate a solution? For example? Do you think your ability of generating solutions was changed? Why? For example?
8. How did you make decision of selecting a solution? For example? Do you think your ability of making decision was changed? Why? For example?

9. How did you implement and verify the selected solutions? For example? Do you think your ability of implementing and verifying solutions was changed? Why? For example?
10. Do you think S-T/S-S/S-R interactions were helpful for the development of confidence/abilities mentioned above? Why? For example?

Part six: Computational thinking skills

1. Do you know what is computational thinking skills? Can you describe this term based on your own experiences?
2. Can you classify the kinds of problems you encountered when you were programming? For example?
3. How did you formulate a problem before programming? For example? Do you think your ability of formulating problem before programming was changed? Why? For example?
4. How did you abstract a problem before programming? For example? Do you think your ability of abstraction was changed? Why? For example?
5. Do you think the programs you wrote were logical? Why? For example? Do you think your ability of logical thinking was changed? Why? For example?
6. Do you think you have used some programming logics (such as 'if-then', 'repeat', etc.) in your programs? For example? Do you think your ability of using these logics was changed? Why? For example?
7. Do you think you have debugged your programs? For example? Do you think your ability of debugging was changed? Why? For example?
8. Do you think S-T/S-S/S-R interactions were helpful for the development of abilities mentioned above? Why? For example?

Appendix H: Observation Form (for the Researcher)

DATE: TIME: CLASS: STUDENT ATTENDANCE: TEACHER: LESSON TASK:		
S-T interactions		
	Examples (Time, participants, behaviors)	Relation to PS/CT
Modeling		
Coaching		
Scaffolding		
Articulation		
Reflection		
Exploration		
S-S interactions		
	Examples (Time, participants, behaviors)	Relation to PS/CT
Discussion		
Observational learning		
Off-task interactions		
S-R interactions		
	Examples (Time, participants, behaviors)	Relation to PS/CT
Building		
Programming		
Observing		
Debugging		

Appendix I: Details of Every Lessons

Lesson 1. Introduction to the Intelligent Brick and Scratch

Purposes:

- To learn the basic functions of the Intelligent Brick, including running program, intra-system test, data revision, and system setting.
- To know how to download the program from computer to the Intelligent Brick.
- To know how to test and set the motors with Intelligent Brick.
- To learn the basic knowledge of Scratch.

Contents:

- Four basic functions of the Intelligent Brick: running program, intra-system test, data revision, system setting.
- Using Intelligent Brick to test and set the motors.
- Introduction to the Scratch.

Assessment:

Using Scratch to write a simple program for controlling the motors; download to the Intelligent Brick.

Lesson 2. Make your vehicle 1 (Moving forward)

Purposes:

- To learn the structure of a vehicle.
- To learn how to build a vehicle by robot kit elements.
- To learn how to write a Scratch program for making the vehicle move forward.

Contents:

- The structure of a vehicle.
- Elements for building a vehicle.
- The “electronic module” block in Scratch and its usage (Motor).

Assessment:

Build your own vehicle and make it move forward (or as you wish).

Lesson 3. Make your vehicle 2 (Go and return)

Purposes:

- To learn how to write a Scratch program for making the vehicle move forward for several seconds and return back.

Contents:

- The use of the “Control” block in Scratch (Repeat and Wait).

Assessment:

Build your own vehicle and make it move forward for several seconds (depends on you) and return back.

Lesson 4. Make your vehicle 3 (Touch and return)

Purpose:

- To learn how to build a vehicle with a touch sensor.
- To learn how to write a Scratch program for making the vehicle return back as long as touching a wall.

Contents:

- The use of a touch sensor.
- The setup and connection of a touch sensor.
- The use of “Control” block in Scratch (Repeat, Wait, and If-Then).
- The use of “Operators” block in Scratch.
- The “electronic module” block in Scratch (Motor and touch sensor).

Assessment:

Make a vehicle which can move and return when it touches a wall.

Lesson 5. Make your vehicle 4 (Line following)

Purposes:

- To learn to build a vehicle with a colour sensor.
- To learn how to write a Scratch program for making the vehicle move following the black line.

Contents:

- The use of a light sensor.
- The setup and connection of light sensor.
- The use of “Control” block in Scratch (Repeat, Wait, and If-Then).
- The use of “Operators” block in Scratch.

Assessment:

Make a vehicle which can move following the black line.

Lesson 6. Make your vehicle 5 (An off-road vehicle)

Purposes:

- To learn the structure of an off-road vehicle.
- To learn how to build an off-road vehicle.
- To learn how to write a Scratch program for making the off-load vehicle move forward and turn left (or turn right).

Contents:

- The structure of an off-road vehicle and the use of the caterpillar.
- How to make the vehicle turn to the left/right by programming.

Assessment:

Make an off-road vehicle which can turn to the left/right.

Lesson 7. Make your fan

Purposes:

- To learn the structure of a fan.
- To learn how to make a fan by robot kit elements.
- To learn how to write a Scratch program for turning on/off the fan when pressing the button.

Contents:

- The structure of a fan and elements for building a fan.
- The setup and connection of a touch sensor.
- Using the “electronic module” to control a motor.
- Using the “electronic module” to read the data from the touch sensor.

Assessment:

Make a fan which can be turned on and turned off by pressing the button.

Lesson 8. Make your toy gyro

Purposes:

- To learn the structure of a toy gyro.
- To learn how to make a toy gyro by robot kit elements.
- To learn how to write a Scratch program for “spinning” the gyro.

Contents:

- The structure of a toy gyro.
- Elements for building a gyro.
- Using the “electronic module” to control a motor.

Assessments:

Build your own gyro and make it spin.

Lesson 9. A “walking” robot

Purpose:

- To learn the structure of a “walking” robot.
- To learn how to make it by robot kit elements.
- To learn how to write a Scratch program for making it walks forward and stop.

Contents:

- The structure of a “walking” robot.
- The use of gears for making the robot “walk” like a human.
- Using the “electronic module” to control a motor.

Assessment:

Make a “walking” robot which can walk like a human.

Lesson 10. A Dagao machine (i.e., in Chinese “打糕机”, a machine for making a Chinese rice cake)

Purpose:

- To learn the structure of a Dagao machine.
- To learn how to make it by robot kit elements.
- To learn how to write a Scratch program for turning it on/off.

Contents:

- The structure of a Dagao machine.
- Using the “electronic module” to control a motor.

Assessment:

Make a Dagao machine with one motor and several gears.

Lesson 11. A robot after you

Purpose:

- To learn how to make a robot which can follow your steps.
- To learn the use of an ultrasonic sensor.
- To learn how to write a Scratch program for allowing the robot: a) follows your steps when you are walking forward; b) stop when you stop.

Contents:

- The structure of this kind of robot. - The use of an ultrasonic sensor. - Using “If-Then” of the “Control” block in Scratch. Assessment: Make a robot after you which can follow your steps.
Lesson 12. A simple football robot Purposes: - To learn the basic structure of a football robot. - To learn how to detect a lighting football with the colour sensor. - To learn how to write a Scratch program for a football robot which can: a) detect the football; b) hit the football. Contents: - The basic structure of a football robot. - The use of a colour sensor in a football robot. - Using “Repeat” and “If-Then” of the “Control” block for detecting and hitting the lighting football. Assessment: Using only one colour sensor to make a simple football robot which can detect and hit the light football.