Rodrigo, M. M. T. et al. (Eds.) (2021). Proceedings of the 29th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

From Mathematical Thinking to Computational Thinking: Use Scratch Programming to Teach Concepts of Prime and Composite Numbers

Siu Cheung KONG[a*] & Wai Ying KWOK[b]
[a]*Department of Mathematics and Information Technology, The Education University of Hong Kong, Hong Kong*
[b]*Centre for Learning, Teaching and Technology, The Education University of Hong Kong, Hong Kong*
*sckong@eduhk.hk

Abstract: This study pioneered the pedagogical use of Scratch programming to support Grade 6 students to cross from mathematical thinking to computational thinking (CT) in mathematics classrooms. A Scratch-based pedagogical innovation was designed to expose students to the pedagogy "To Play, To Think, To Code" with two Scratch apps and five Scratch activity worksheets to explore, think about, apply and consolidate the target mathematical concepts through Scratch programming. An eight-lesson teaching in 320 minutes was trialed in 15 selected Grade 6 classes involving 324 students from seven primary schools in Hong Kong. From the pre-post-tests, the pedagogical innovation successfully supported students to make statistically significant growth in understanding all five topic-specific mathematical concepts and all five target CT concepts. From the questionnaire surveys, students demonstrated a high level of awareness of the two target CT practices, and a positive perception of CT development for their own good. From the focus group interviews, students confirmed the effectiveness of and expressed a satisfaction with the pedagogy for mathematics learning and CT development through coding. The positive results of this study confirm the potential of the pedagogical innovation which integrates CT education with subject-specific curriculum delivery for an effective development of both subject knowledge and CT competency among primary school students. Implications for the flow and scope of future integration of subject lessons with coding activities are discussed.

Keywords: Computational thinking, mathematical thinking, primary schools, prime and composite numbers, Scratch programming

1. Introduction and Background of Study

Computational thinking (CT) is advocated essential for everyone to succeed in the digitalized society (Grover & Pea, 2013; Shute, Sun, & Asbell-Clarke, 2017). Wing (2006, p.33) defines CT as a thinking process for "solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science". CT has three dimensions: CT concepts – the common concepts used in programming such as sequence, conditionals, repetition; CT practices – the process of programming such as iterative and incremental, abstracting and modularizing, testing and debugging; and CT perspectives – students' understandings of themselves and the technological world (Brennan & Resnick, 2012; Rodríguez-Martínez, González-Calero, & Sáez-López, 2020). Educators realize the necessity to develop CT among students, and so the importance of CT education in school curriculum. The advent of block-based programming environments such as Scratch creates potential to introduce CT activities in subject teaching at primary grades for co-developing subject knowledge and CT (Gadanidis, 2015; Lee, Grover, Martin, Pillai, & Malyn-Smith, 2020). This study tapped into such pedagogical potential – to pioneer the use of Scratch programming among senior primary students to co-develop mathematical concepts of "Prime and Composite Numbers" and CT competency.

The "Number" strand is a central component in primary mathematics curriculum; and "Prime and Composite Numbers" is a main topic in the "Number" strand (Ustunsoy, Ozdemir, & Unal, 2011;

Zazkis & Zazkis, 2014). This topic has three vital knowledge points – 1) any natural number greater than "1" is either a prime or a composite; 2) when a number is represented as a product, it is a composite number unless the factors are "1" and a prime number; and 3) composite numbers have a unique prime decomposition (Dickerson & Pitman, 2016; Ustunsoy et al., 2011). There is a big challenge among students when they learn this mathematical topic – an inadequacy of the concept that a prime number has exactly two factors, not more and not less (Mohyuddin & Khalil, 2016; Ustunsoy et al., 2011). Researchers such as Dickerson and Pitman (2016) and Zazkis and Zazkis (2014) advocate the best way to master the knowledge points and address the learning challenge abovementioned is to develop the concept of using the number of factors of the given natural numbers for categorization – prime numbers have only two factors; while composite numbers have more than two factors.

The block-based programming environment Scratch is popularly used in subject classrooms in primary education (Benton, Hoyles, Kalas, & Noss, 2017; Rodríguez-Martínez et al., 2020). Its intuitive interface-design allows children to make simple actions on dragging, dropping, and combining code blocks to easily create programs and immediately observe the programming outcomes (Calder, 2019; Gadanidis, 2015). Frameworks by Brennan and Resnick (2012) and Grover et al. (2017) are widely referred for integrating CT education into subject curriculum via Scratch programming environment – wherein the coding products serve as computational manipulatives which conceptually align with the traditional notion of educational manipulatives (Calder, 2019; Rodríguez-Martínez et al., 2020).

There is a natural fit to integrate CT education into mathematics curriculum delivery, due to a shared logical structure in the developmental process of algorithmic thinking between mathematical thinking and CT (Gadanidis, 2015; Pérez, 2018). Algorithmic thinking emphasizes the use of a series of ordered steps to solve problems (Rich, Yadav, & Schwarz, 2019; Yadav, Stephenson, & Hong, 2017). There is a complementary connection in students' development between mathematical thinking and CT – to link up abilities of pattern generalization and abstraction (Pérez, 2018; Rich, Spaepen, Strickland, & Moran, 2020). The ability of pattern generalization sets to analyze algorithmic representations to discover regularities within that set of algorithmic representations. Then, the ability of abstraction sets to translate the discovered regularities into a concise and precise mathematical formula, and create an abstraction realizing mathematical formula to be a programmable solution to solve contextual problems automatically. This developmental process is important yet difficult for young students to achieve.

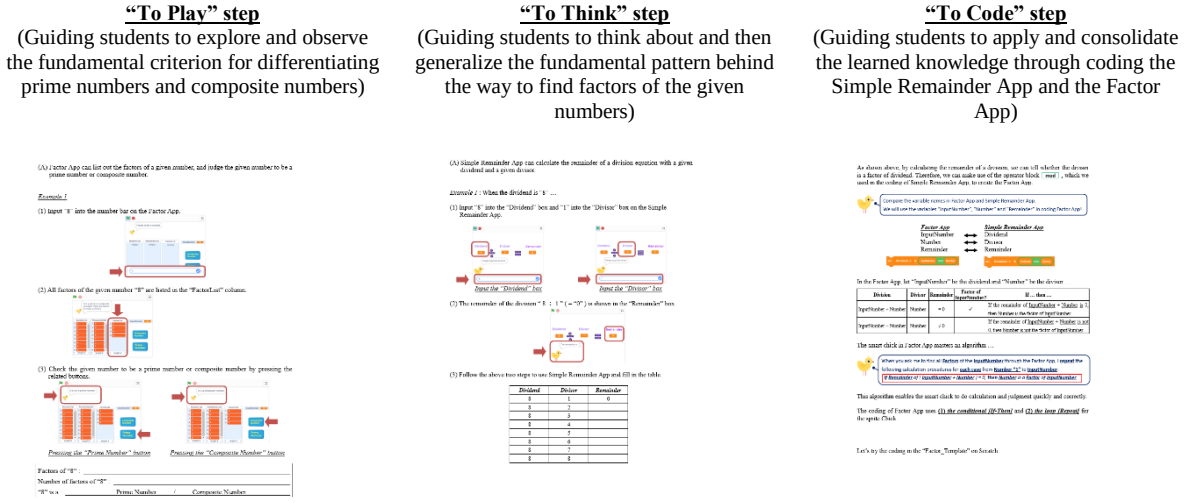
In mathematics subject, there are three main criteria for the pedagogical designs for embedding CT education in subject curriculum. First, the pedagogical designs give students enough chances to work on the selected coding products to construct subject knowledge and stimulate their interest in coding (Calder, 2019; Rodríguez-Martínez et al., 2020). Second, the pedagogical designs give students enough chances to apply subject knowledge in thinking about programming solutions for solving problems in subject-specific contexts (Chiang & Qin, 2018; Rodríguez-Martínez et al., 2020). Third, the pedagogical designs give students enough chances to consolidate subject knowledge in generating coding products for solving subject-specific problems (Benton et al., 2017; Chiang & Qin, 2018).

2. The Study: Research Design and Evaluation Methods

This study pioneered the research on developing both subject knowledge and CT competency through block-based programming activities in subject classrooms. It aimed to innovate a pedagogical design which engages students in Scratch programming for developing the important knowledge of using the number of factors to formulate the concepts of prime numbers (having only two factors) and composite numbers (having more than two factors); and at the same time the competency of five CT concepts ("sequences", "events", "conditionals", "repetition", and "operators"), two CT practices ("iterative and incremental" and "testing and debugging"), and one CT perspective ("ability to connect") – as in the CT frameworks by Brennan and Resnick (2012) and Rodríguez-Martínez et al. (2020).

The pedagogical innovation consisted of a three-step pedagogy "To Play, To Think, To Code" and a Scratch programming environment with two Scratch apps for stimulating students to use the number of factors to formulate the concepts of prime and composite numbers, and so to classify the given numbers to be prime or composite numbers. Five Scratch activity worksheets were also designed to support students to learn using the two Scratch apps. Figure 1 illustrates how students were engaged in "playing" the Factor App for inquiry-based learning of prime and composite numbers; "thinking" of

the target concepts through guided-discovery worksheets; and "coding" in Scratch for a programming solution through reusing and refining codes in the Simple Remainder App and the Factor App.



"To Play" step
(Guiding students to explore and observe the fundamental criterion for differentiating prime numbers and composite numbers)

Students used the Factor App to explore different numbers for a judgment to be "prime numbers" and "composite numbers".

"To Think" step
(Guiding students to think about and then generalize the fundamental pattern behind the way to find factors of the given numbers)

Students used the Simple Remainder App to calculate and tabulate the remainders of the given division-equations.

"To Code" step
(Guiding students to apply and consolidate the learned knowledge through coding the Simple Remainder App and the Factor App)

Students made before-coding reflection on the learned knowledge of the relationship between "Remainders" and "Factors" for finding factors of the given numbers.

Figure 1. Sample Questions in Scratch Activity Worksheets in the "To Play, To Think, To Code" Steps.

This study had 15 Grade 6 classes from seven Hong Kong primary schools – involving a total of 324 students – for a consented participation (see Table 1). The students had some background knowledge of the target mathematical topic, as they learned this topic previously in Grade 4 or Grade 5 mathematics curriculum. The students had some experience in programming before participating in this study. The mathematics teachers of these 15 participating classes trialed the pedagogical innovation on a class-specific basis. Before the trial teaching, the participating teachers completed a four-hour training workshop which prepared them well for a sound recognition of the rationale of learning through coding in local mathematics curriculum; a ready implementation of the "To Play, To Think, To Code" pedagogy in mathematics classroom; and a confident integration of the two Scratch apps and the five Scratch activity worksheets into topic-specific lessons. This study focused on two research questions: (1) What did the students achieve in developing mathematical concepts and CT under the pedagogical innovation? (2) How did the students perceive the pedagogical innovation for developing CT in mathematics classrooms? Three methods were adopted for evaluating the pedagogical innovation, adopting the research instruments developed by the research team with backgrounds in mathematics, education, and computer sciences.

Table 1. *Profile of Students Participated in This Study.*

	School A	School B	School C	School D	School E	School F	School G
No. of students	28	82	68	75	32	12	27
No. of classes	1	3	4	4	1	1	1
Boys : Girls	16:12	37:45	36:32	45:30	17:15	8:4	12:15
Mean age (years)	11.11	10.85	10.93	10.94	10.84	10.92	11.00

Firstly, the pre-post-tests were conducted at the beginning and the end of the pedagogical innovation to investigate students' achievement in mathematics learning and CT development. The test papers contained 15 questions: four questions on testing the concept of composite and prime numbers, one on the concept of "1" is neither prime number nor composite number, two on the relationship between composite numbers and multiples, three on finding prime numbers by enumeration, one on finding factors of a number, and four on CT concepts including "operator", "repetition", "events &

conditionals" and "sequence". A statistical comparison of students' pre-test and post-test scores was conducted with the assistance of SPSS software. The Cronbach's alpha reliability coefficients for the pre-test and post-test are 0.785 and 0.754 respectively.

Secondly, the pre-post-surveys were conducted at the beginning and the end of the pedagogical innovation to investigate students' perception of developing CT in mathematics classrooms. The questionnaire contained five 5-point Likert scale questions, of which three questions on the building of awareness, interest and confidence in programming, and two on the development of CT practices including "iterative and incremental" and "testing and debugging". The mean rating for each question and the corresponding standard deviation were then calculated. The Cronbach's alpha reliability coefficients for the pre-survey and post-survey are 0.827 and 0.861 respectively.

Thirdly, focus group interviews were conducted at the end of the pedagogical innovation to investigate students' perception of the pedagogical innovation for developing CT. A total of 25 students were randomly selected from the seven participating schools, with each focus group consisting of three to five students. The student respondents were asked about how they perceived the help from the pedagogical innovation in their development of mathematical concepts and CT competency, the enjoyment in and satisfaction with the pedagogy for mathematics learning through coding, and the challenges in and recommendations for mathematics lessons integrated with coding activities. All the interview content was transcribed and systematically summarized.

3. Results and Discussion

3.1 Students' Achievement in Developing Mathematical Concepts and CT

The pre-post-tests found that the pedagogical innovation effectively supported students to develop mathematical concepts on prime and composite numbers (see Table 2). The students had a statistically significant increase in the post-test scores for the question items on all topic-specific knowledge points.

Table 2. *Students' Achievement in Developing Concepts of Prime and Composite Numbers Before and After the Pedagogical Innovation (N = 324).*

Question items			Pre-test scores		Post-test scores		
Topic concepts	No. of items	Max. scores	Mean	(SD)	Mean	(SD)	t-test
A. Concept of Factor	7	14	6.91	(3.16)	9.00	(3.17)	12.53***
(1) Concept of composite and prime numbers	4	8	4.45	(2.18)	5.87	(2.12)	11.99***
(2) Concept of "1" is neither prime number nor composite number	1	2	1.18	(0.62)	1.46	(0.57)	6.87***
(3) Exploring the relationship between composite numbers and multiples	2	4	1.28	(1.06)	1.67	(1.17)	5.47***
B. Finding Factors and Prime Numbers	4	10	8.02	(2.37)	8.67	(1.86)	5.51***
(4) Finding prime numbers by enumeration	3	6	4.69	(1.55)	5.18	(1.21)	6.04***
(5) Finding factors of a number	1	4	3.33	(1.17)	3.48	(1.00)	2.20*
C. Total	11	24	14.93	(4.94)	17.66	(4.46)	12.01***

*$p < 0.05$ ***$p < 0.001$

For learning the concept of factor, the pre-post-test results indicate that students after the pedagogical innovation had a noticeable knowledge gain – with the dimension-specific mean score below the passing score (i.e. half of the maximum score) in the pre-test to finally above the passing score in the post-test. The students were found to improve greatly their concept of composite and prime numbers, and extend their understanding that "1" is neither prime number nor composite number. For

learning the ways of finding factors and prime numbers, students after the pedagogical innovation built on their fairly good knowledge foundation in this dimension for a statistically significant improvement in the knowledge of finding prime numbers by enumeration and finding factors of a number.

The pre-post-tests also found that the pedagogical innovation effectively supported students to develop CT concepts (see Table 3). The students had a statistically significant increase in the post-test scores for the question items on the target CT concepts. The pre-post-test results indicate that students after the pedagogical innovation had a noticeable growth in the mastery of CT concept "operator" – with the mean score below the passing score in the pre-test to finally above the passing score in the post-test. The students were found to maintain their level of mastery of the CT concept "Repetition" throughout the trial teaching – a fair performance of which the mean scores of the related question item in the pre-test and post-test are just-above the passing score, without a statistically significant difference.

Table 3. *Students' Achievement in Developing Concepts of CT Before and After the Pedagogical Innovation (N = 324).*

Question items				Pre-test scores		Post-test scores		
CT concepts		No. of items	Max. scores	Mean	(SD)	Mean	(SD)	t-test
(1)	Operator	1	1	0.45	(0.50)	0.66	(0.48)	5.87***
(2)	Repetition	1	1	0.51	(0.50)	0.50	(0.50)	0.29
(3) & (4)	Events & Conditionals	1	1	0.34	(0.48)	0.44	(0.50)	2.74**
(5)	Sequence	1	1	0.38	(0.49)	0.45	(0.50)	2.27*
Total		4	4	1.68	(1.10)	2.05	(1.16)	5.20***

*$p < 0.05$ **$p < 0.01$ ***$p < 0.001$

3.2 Students' Perception of Developing CT in Mathematics Classrooms

From Table 4, there is no statistical significance in students' perception of the pedagogical innovation for developing CT in mathematics classrooms before and after the trial teaching. As mentioned, the students had some experience in programming before participating in this study. This possibly led the students to keep a high level of agreement with the importance of the step-by-step development of a program (CT practice of "iterative and incremental") and the operability-testing of the program (CT practice of "testing and debugging") before and after the trial teaching.

Table 4. *Results of Students' Questionnaire Survey on the Perception of the Pedagogical Innovation for Developing CT in Mathematics Classrooms (N = 324).*

Items	Pre-survey		Post-survey		t-test
	Mean (1-5) #	(SD)	Mean (1-5) #	(SD)	
I think it is important to test the program to make sure it works.	4.12	(0.96)	4.04	(1.11)	1.14
I think it is important to develop a program step by step.	4.09	(0.98)	4.00	(1.08)	1.30
I think that programming is important in our daily lives.	3.59	(1.03)	3.63	(1.13)	0.64
I am confident that I can write a simple program.	3.38	(1.15)	3.39	(1.17)	0.14
I am interested in learning programming.	3.35	(1.13)	3.32	(1.15)	0.55

#*Note: 1 = "strongly disagree", 2 = "disagree", 3 = "neutral"; 4 = "agree"; 5 = "strongly agree".*

The focus group interviews further confirmed students' positive perception of the pedagogical innovation for developing CT in mathematics classrooms (see Table 5). Echoing with the results of pre-post-tests and surveys, the students confirmed that the mathematically-rich activities on using Scratch apps can effectively support them to master the mathematical concepts on prime and composite numbers, as well as the CT concepts and CT practices targeted at this study. All student respondents indicated that the trial teaching enabled them to master the fundamental concept that when an integer can divide a number without giving a remainder, that integer is the "factor" of the number being divided. A student

respondent illustrated an example that for the number "10", the multiplication equations "1 x 10 = 10" and "2 x 5 = 10" stand; and so the number "10" has "1", "2", "5" and "10" being its four factors. He further elaborated that this number has more than two factors; and so by definition it is a composite number but not a prime number which has two factors only. Nearly a fifth of the student respondents expressed that "1" is an important knowledge point in the trial teaching – they were able to explicate that "1" has only one factor, the number itself; and so this number does not meet the definition of neither a "prime number" nor a "composite number". They pointed out that during the trial teaching many students, without the instruction or request from the teachers, tried to use the apps to check the category of "1" and discover this uniqueness of "1", and in turn developing the correct understanding that "1" is neither a "prime number" nor a "composite number". Two student respondents indicated that they used the apps to extend their learning exploration of the number "0" which is seldomly covered by traditional mathematics textbook; and noticed that "0" is a special number which neither prime nor composite. This finding implies the need for teachers to discuss with students about the uniqueness of the special numbers "1" and "0". Nearly three quarters of the student respondents indicated that the trial teaching, comparing with the typical teaching approach, can foster them to think more about the mathematical concepts behind the process of classifying prime and composite numbers through checking the number of factors of the given numbers. These student respondents pointed out that they seldomly think about the meaning and purpose of each calculation step. They appreciated the coding activities dissected each calculation step to give a clear visualization of the calculation process; and this fostered them to widen their thinking angles to look into what and why each of those calculation steps is necessary.

Table 5. *Feedback from Students' Focus Group Interviews on the Perception of the Pedagogical Innovation for Developing CT (N = 25).*

Major interview feedback
Help in the development of mathematical concepts and CT
• The apps in the trial teaching supported students to quickly identify all factors of the given numbers and accurately classify the given numbers into prime numbers and composite numbers. • The trial teaching enabled students to explore some very large numbers and the special numbers "1" and "0" that are seldomly covered by traditional mathematics textbook, as the apps were convenient to use for finding factors of the given numbers quickly and accurately. • The apps in the trial teaching impressively saved students' time to manually find factors of the given numbers; and more lesson time can be arranged for student-student and student-teacher interactions to exchange topic-specific concepts. • The steps in the coding activities served as a clear guidance of classifying primes and composites through checking the number of factors of the given numbers. Students were well supported to better understand the concepts behind the process of identifying primes and composites. • The trial teaching motivated students' extra efforts to refine and debug the codes for the apps after class time to improve existing features and add new features, such as to make the apps able to process negative integers. Students got a great sense of achievement and confidence when the coding products can operate as intended. • This learning experience inspired students to make the best efforts to try different alternatives for solving problems in coding and daily life. Students made many errors at the beginning of the coding process; and they were willing and committed to carefully check the coding outcomes, review their codes, correct the sequence, and vary the parameters of the command blocks for many trials to get the intended coding outcomes.
Enjoyment in and satisfaction with the pedagogy for mathematics learning through coding
• Learning mathematics through coding is considered very interesting and meaningful. • The coding tasks were the most popular part during the trial mathematics lessons. • The apps in the trial teaching were so attractive and interesting to stimulate learning motivation. • There was a great enjoyment in the coding activities which are new and challenging. • Coding tasks were highly satisfied, with a great sense of achievement after coding successfully. • The steps in the activity worksheets were confirmed to be clearly and comprehensively stated. The questions inside were considered effective to guide students to progressively deepen their knowledge of the target topic as well as coding.

Challenges in and recommendations for mathematics lessons integrated with coding activities
• Some technical problems occurred in the coding lessons, as not every student was familiar with Scratch programming. • Learning diversity existed among students in the coding lessons - some students mastered coding quickly while some not. • Each student should have two computing devices during the trial lessons: one for viewing teachers' lecturing, and the other one for coding along with teachers' demonstration. • Pre-training of Scratch coding should be arranged before the trial lessons for familiarization. • Other mathematical topics suitable for learning through coding include "Percentage", "3D Shapes", "Addition and Subtraction", and "Equations". • The approach of learning through coding can be extended to students at lower grades and to other subjects, such as the topic of "Storytelling" in language subjects at Grade 1; the learning of vocabulary in English Language subject; and the learning of the classification of animals in General Studies subject.

Nearly all student respondents appreciated that the coding activities can guide them to discover a series of calculation steps that teachers seldomly mention in typical mathematics classroom when teaching the target topic. This allowed them to develop mathematical concepts and coding knowledge at the same time. A student respondent further indicated his impression that the coding activities equipped them with knowledge of some operators in Scratch coding, such as the [mod] operator-block (i.e. "modulo") for division equations. Nearly 45% of the student respondents demonstrated a high awareness of debugging in the coding process, as reflected in the response that it is a must to order codes in a correct sequence during the coding process, for ensuring the apps are able to operate as intended. Four of them reflected that there were many errors made at the beginning of the coding process, and they were willing and committed to carefully check the coding outcomes, review their codes, correct the sequence, and vary the parameters of the command blocks for many trials to get the intended coding outcomes. They impressively linked the debugging step in coding with the procedures-check in mathematical calculation – that both needed to be attentive to the minute steps in order to give the intended outcomes. This learning experience inspired them to make the best efforts to try different alternatives for solving problems in coding and daily life.

The students also enjoyed and felt satisfied with the pedagogy for mathematics learning through coding. Nearly all student respondents indicated that they liked and committed to use the apps for a quick check on the number of all factors for the given numbers. They preferred to have this innovative approach in mathematics classrooms. Around 45% of the student respondents confirmed that the learning approach in the trial teaching can effectively combine mathematics learning with CT development. They were satisfied with the benefits of developing both mathematical concepts and CT competency in one subject lesson. Nearly 30% of the student respondents confirmed that it was interesting to use the apps for supporting the learning of the target topic. They considered lessons in the trial teaching were less boring, comparing the traditional mathematics classrooms, as they were allowed to play apps during the learning process. Two of them further indicated their extra efforts to refine and debug the codes for the apps after class time for improving existing and add new features, such as to make the apps able to process negative integers. They got a great sense of achievement and confidence when the coding products can operate as intended.

There was feedback on two challenges and four recommendations for mathematics lessons integrated with coding activities. Two student respondents explained that it was their first time to do Scratch coding and they were unfamiliar with Scratch programming environment. This led to two main challenges in the pedagogical innovation: some technical problems occurred in the coding lessons; and some students lagged behind in the progress of coding activities. In this regard, the student respondents indicated their expectation for pre-training of Scratch coding before the trial lessons, followed by the existing appropriate approach of first teaching about the basic concepts of the target mathematical topic, and then teaching about the knowledge and skills of coding. They stressed that it is good for them to have two computing devices during the trial lessons, in which they use one device to view teachers' lecturing and the other device to code along with teachers' demonstration. Four student respondents suggested that "Percentage", "3D Shapes", "Addition and Subtraction", and "Equations" are mathematical topics suitable for learning through coding. The other three student respondents further

suggested that it is possible to extend the approach of learning through coding to support students at lower grades; or to support the learning and teaching in other subject topics. One of these student respondents made suggestion on the topic of "Storytelling" in language subjects at Grade 1. The other student respondent made suggestion on the learning of vocabulary in English Language subject; and the learning of the classification of animals in General Studies subject.

3.3 Interaction between Mathematical Thinking and Computational Thinking in Student Learning under the Pedagogical Innovation

From the results of this study, the pedagogical innovation with the three-step "To Play, To Think, To Code" approach can benefit students in the learning of mathematical and computational domains.

The first half of the pedagogical innovation – the mathematically-rich activities on using Scratch apps – exposes students to the interaction from mathematical thinking to CT. This part is attested to be effective, as reflected by students' feedback from focus group interviews on appreciating the support from Scratch apps for them to dissect and visualize each calculation step of finding factors; illustrating the procedures and justification for finding factors of a number such as "10"; as well as indicating the extension of self-initiative to explore the categories and discover the uniqueness of the special numbers "1" and "0". These results also imply that the mathematically-rich activities in the pedagogical innovation are potential to help students address their common topic-specific learning challenge as suggested by Mohyuddin and Khalil (2016) and Ustunsoy et al. (2011) – as students under the pedagogical innovation can understand "1" is not a prime number and firmly grasp the concept that a prime number has exactly two factors.

The second half of the pedagogical innovation – the computationally-rich activities on programming Scratch apps – exposes students to the interaction from CT to mathematical thinking. This part is also attested to be effective, with evidence from focus group interviews in which students explicated the grasp of knowledge about Scratch coding blocks for mathematical operations such as [mod] operator-block in division equations; indicated the awareness of outcomes check, codes review, sequence correctness, and parameters variation during coding process; as well as linked the debugging step in coding with procedure-check in mathematical calculation. These results also imply that the computationally-rich activities in the pedagogical innovation echo with the advocacy from Gadanidis (2015) and Pérez (2018) to be a potential approach to a fitting integration of CT education into mathematics curriculum delivery through linking up the developmental processes of algorithmic thinking between mathematical thinking and CT.

The interaction from mathematical thinking to CT achieved by the students can be attributed to the four types of engagement in the mathematically-rich activities. In the first half of the pedagogical innovation, students first developed *(i) mathematical thinking* through working with Scratch activity worksheets to tabulate and observe the pattern that "If Remainder of Dividend ÷ Divisor is / is not 0, then Divisor is / is not a Factor of Dividend" among the given division-equations. Students based on the tabulation results to generalize the pattern that when a given number is divided by a divisor in the range of 1 to the given number, a "zero" remainder in the related division-equation means the divisor is the factor of that given number. Next, students developed *(ii) computational thinking* to think about the possibility to write a computing program to automatically solve the problem of finding factors of a given number; and the need for the computing program to get a feature to store the factors of the given number. Students were subsequently guided to think about the need to come up with an algorithm to find out all factors of a given number. Accordingly, students were led to link up mathematical thinking with CT through the step of *(iii) abstraction* (i.e. making an abstraction of the mathematical pattern that when a given number is divided by a divisor in the range of 1 to the given number, a "zero" remainder in the related division-equation means the divisor is the factor of that given number); and the step of *(iv) algorithmic thinking* (i.e. setting the algorithm with the necessary variables for the computing program, based on their abstraction of the tabulated mathematical pattern of "If Remainder of Dividend ÷ Divisor is / is not 0, then Divisor is / is not a Factor of Dividend").

The interaction from CT to mathematical thinking achieved by the students can be attributed to the three types of engagement in the computationally-rich activities. In the second half of the pedagogical innovation, students progressively developed *(v) CT concepts* and *(vi) CT practices* through implementing the algorithm in the Scratch programming environment by the step of reusing

the codes from the Simple Remainder App to refine the Factor App; and the step of testing if their Scratch program can correctly list out all factors of the given numbers and judge the given numbers as prime numbers or composite numbers. Students' development of CT concepts covered "sequences", "events" [When …], "conditionals [If-Then] and [If-Then-Else]", "repetition [Repeat]", "operators [Mod], [=], [>] and [<]"; and their development of CT practice covered "reusing and remixing", "iterative and incremental" and "testing and debugging". Finally, students were led to develop *(vii) CT perspectives* through connecting the mathematical task on decomposing relatively large prime numbers with the significant role of prime numbers in information security in digital communication – this step demonstrated the CT perspectives of "ability to connect".

The mathematically-rich activities and the computationally-rich activities in the pedagogical innovation on the whole support students on progressing to an interaction between mathematical thinking and CT through linking up the abilities of pattern generalization and abstraction. The series of "To Code" activities started at leading students to develop and demonstrate CT competency through Scratch programming; and ended with fostering students to apply and consolidate their mathematical understanding developed through the "To Play" and "To Think" activities. Apart from these activities, teachers should reflect with students about the uniqueness of the numbers "1" and "0" on top of natural numbers greater than "1" in the context of prime and composite numbers (Kong, 2019).

4. Conclusion and Future Direction

This study developed and implemented a pedagogical innovation which used Scratch programming to support 324 students from 15 selected Grade 6 classes in seven primary schools at Hong Kong to co-develop mathematical concepts and CT competency in subject classrooms. Under an eight-lesson teaching supported by the pedagogy "To Play, To Think, To Code" with two Scratch apps and five Scratch activity worksheets, students in this Scratch-based pedagogical innovation explored, thought about, applied and consolidated the mathematical concepts of prime and composite numbers through Scratch programming; in which also went through the development and application of CT concepts, CT practices, and CT perspectives. The pre-post-tests confirmed the effective support from the pedagogical innovation for students to significantly enhance their understanding of mathematical knowledge about the concepts of factors, composite numbers and prime numbers, and the ways of finding factors and prime numbers; as well as CT concepts of "operator", "events & conditionals" and "sequence". The questionnaire surveys confirmed that the pedagogical innovation successfully fostered students to be highly aware of CT practices of "iterative and incremental" and "testing and debugging". The focus group interviews confirmed that students positively perceived the pedagogical innovation to be an effective and satisfying pedagogy for mathematics learning and CT development through coding.

The evidence found in this study implies that the pedagogical innovation is potential to effectively support primary school students to co-develop mathematical concepts and CT competency. It is promising to first engage students in mathematically-rich activities on using Scratch apps to explore the concepts of prime and composite numbers and generalize the pattern of finding factors of the given numbers; and then guide students to cross from mathematical thinking to CT for making abstraction and algorithmic thinking – to translate the generalized pattern into a mathematical formula in pseudocodes – for a programmable solution which automatically finds factors of the given numbers and categorizes prime and composite numbers; and finally engage students in the computationally-rich activities on programming Scratch apps – in which they develop and apply CT concepts, CT practices, and CT perspectives when generating a programmable outcome for the target solution-automation.

The recommendations collected from this study imply the potential to expand the application scope of the pedagogical innovation to other mathematical topics, other grades, and/or other subjects. One of the possible future directions will be the pedagogical innovation for using Scratch programming in Grade 4 English Language classrooms to support students to develop building blocks for the learning and teaching of locations and directions. The future research will try to address the need of two computing devices for each student to view lecturing and perform coding, for a smoother lesson flow. This study had a limitation of no control group involved for evaluating the effectiveness of the pedagogical innovation. Future research will also try to arrange a control group for evaluation purposes.

Acknowledgements

The authors would like to acknowledge the support of the project "Use Coding as a Pedagogy for Teaching Subject Knowledge in Mathematics, General Studies and English Language and Fostering Computational Thinking" funded by the HKSAR Quality Education Fund.

References

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digital Experiences in Mathematics Education*, *3*(2), 115-138.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *2012 Annual Meeting of the American Educational Research Association (AERA'12)*, Canada.

Calder, N. (2019). Using Scratch to facilitate mathematical thinking. *Waikato Journal of Education*, *23*(2), 43-58.

Chiang, F.-K., & Qin, L. (2018). A pilot study to assess the impacts of game-based construction learning, using Scratch, on students' multi-step equation-solving performance. *Interactive Learning Environments*, *26*(6), 803-814.

Dickerson, D. S., & Pitman, D. J. (2016). An examination of college mathematics majors' understandings of their own written definitions. *Journal of Mathematical Behavior*, *41*, 1-9.

Gadanidis, G. (2015). Coding as a Trojan Horse for mathematics education reform. *Journal of Computers in Mathematics and Science Teaching*, *34*(2), 155-173.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43.

Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education*, *17*(3), 14.

Kong, S. C. (2019). Learning composite and prime numbers through developing an app: An example of computational thinking development through primary mathematics learning. In S. C. Kong & H. Abelson (Eds.), *Computational thinking education* (pp. 145-166). Singapore: SpringerOpen.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, *29*, 1-8.

Mohyuddin, R. G., & Khalil, U. (2016). Misconceptions of students in learning mathematics at primary level. *Bulletin of Education and Research*, *38*(1), 133-162.

Pérez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education*, *49*(4), 424-461.

Polly, D. (2011). Examining how the enactment of TPACK varies across grade levels in mathematics. *Journal of Computers in Mathematics and Science Teaching*, *30*(1), 37-59.

Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2020). Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments*, *28*(3), 272-283.

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education*, *27*(2), 165-205.

Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interactive Learning Environments*, *28*(3), 316-327.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158.

Ustunsoy, S., Ozdemir, A. S., & Unal, H. (2011). The investigation of student approach to problem solving about some topics of Number Theory. *Procedia - Social and Behavioral Sciences*, *15*, 3422-3425.

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

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, *60*(4), 55-62.

Zazkis, R., & Zazkis, D. (2014). Script writing in the mathematics classroom: Imaginary conversations on the structure of numbers. *Research in Mathematics Education*, *16*(1), 54-70.