A Project entitled

Effectiveness of Hour of Code to Enhance Computational Thinking of Hong Kong Junior Secondary Students - Echo with STEM Education in Hong Kong

Submitted by

Lam Tik

submitted to The Education University of Hong Kong

for the degree of *Bachelor of Education (Honours) (Secondary) – Information & Communication Technology*

in *May 2017*

Declaration

I, *Lam Tik*, declare that this research report represents my own work under the supervision of *Dr. So Wing Wah*, and that it has not been submitted previously for examination to any tertiary institution.

Signed ______________________

Lam Tik

12th May 2017

Table of Contents

Introduction

Curriculum Development Council (CDC) of Education Bureau has released the latest document named "Promotion of STEM Education – Unleashing Potential in Innovation" at the early November of 2015. The document focuses on promoting STEM Education in Hong Kong, to nurture Hong Kong students' ability on cross-discipline knowledge application, and training students' collaboration and problem solving skills via activities (Curriculum Development Council, 2015). At the year before, Facebook CEO Mark Zuckerberg and his teammates visited a high school in America and encourage the students to step into STEM fields (Chea, 2014). They hope more people, especially girls, to explore coding in STEM fields.

In line with Mark Zuckerberg and his teammates, President of United States Barack Obama and Prime Minister of Singapore Lee Hsien Loong has given a speech via YouTube and released source code of a Sudoku solver via Facebook respectively, to ask their citizens learn coding (Code.org, 2013; Lee, 2015). There is a trend that promoting coding to teenagers even children over worldwide. Instead, IBM China/Hong Kong Limited (2013) has produced a consultancy service report and encourages Hong Kong students to do programming. IBM also states that the main goal of programming is not to increase the coding ability of Hong Kong students, but "to encourage their active exploitation of technology for problem solving" (2013, p. 82), which is echo with the vision of STEM Education promoted by CDC.

Literature Review

Relationships between Coding and STEM Education

The Royal Society stated that Computer Science able to develop key thinking skills, which are logical reasoning, modelling, abstraction and problem-solving (Furber, 2012, p. 29). Computer Science also becomes a tool to unleash students' creativity and invading the skills into other disciplines. In general, the set of mentioned key thinking skills and the integral knowledge application in every other discipline is called "Computational Thinking".

Wing (2006) firstly introduced the concept of Computational Thinking into education. She proposed to implement Computational Thinking into other disciplines and leading other educators and cognitive learning scientists to find out the most effective learning (teaching) approach on Computational Thinking to students. Computational Thinking has 6 main characteristics, which is able to formulate problems into computer programs to find out solution, analyze and organize data logically, apply abstractions such as models and simulations to represent data, automate solutions through algorithmic thinking, find the most efficient and possible solution to the problems, and implement problem solving skills from one problem into others (Stephenson & Barr, 2011, p.3).

Recent years, several organizations and representatives (Barr, Harrison & Conery, 2011; Barr & Stephenson, 2011; Google, 2016; Weintrop, Beheshti, Horn, Orton, Jona, Trouille & Wilensky, 2014) started to implement Computational Thinking into educational curriculum, especially in STEM education. Hu (2011) even claimed that "learning STEM without learning computing is fundamentally inadequate" (p.227). To implement Computational Thinking into classroom in a more effective way, Computer Science Teacher Association (CSTA) and the International Society for Technology in Education (ISTE) (2011) has refined the definition of Computational Thinking into 9 categories[1], and Google (2016) supplements the definition into 11 categories[2].

The characteristics of Computational Thinking lead it to become a crucial element in the educational curriculum reform nowadays. The problem solving skills and cross-discipline knowledge application ability brought by Computational Thinking is noticed by Hong Kong government few years ago. Computational Thinking is also feasible to be one of the key training element in the STEM education promotion since it is echo with the main purpose of the promoted STEM education.

Curriculum Integration of Coding among European Countries and Hong Kong

Although it is feasible to enhance students' Computational Thinking via coding, the worldwide curriculum implementation is also a referable indicator to what stage (grade) we should implement, where we should start to implement, and how deep the implementation should get into the curriculum.

[1] The 9 categories are Abstraction, Algorithms & Procedures, Automation, Data Collection, Data Analysis, Data Representation, Problem Decomposition, Parallelization and Simulation.

[2] The 11 categories are Abstraction, Algorithms & Procedures, Automation, Data Collection, Data Analysis, Data Representation, Problem Decomposition, Parallelization, Pattern Generalization, Pattern Recognition and Simulation.

European countries are the first few countries which started to integrate coding into their national education. According to European Schoolnet, there are 13 countries which going to start and started to integrate coding into their upper secondary education; 10 countries will integrate or integrated coding into their primary education, and 3 countries has implemented coding into their entire school education (Balanskat & Engelhardt, 2015, p. 10). Most of the countries implement coding as a part in the computer course; UK treats coding as compulsory in their primary education; Poland, Malta and Slovakia plans to implement or implemented coding into their whole levels of school education, since they believe “computational thinking is a fundamental skill for everyone, not just computer scientists” (Balanskat & Engelhardt, 2015, p. 10-11). The major aim of the implementation of coding by these countries is to develop students’ logical thinking and problem solving skills, and also as a catalyst to attract more students to study Computer Science.

In Hong Kong, Education Bureau (2015) focus on renewing Information & Communication Technology (ICT) in senior secondary and Technology Education Key Learning Area (TEKLA) curriculum in junior secondary. They keep equipping students’ Computational Thinking through coding in the two curricula of both junior secondary and senior secondary. Moreover, Education Bureau started to implement Computational Thinking into General Studies in primary education. They introduce to combine related subject knowledge into Computational Thinking, and using image-based programs like Scratch since the programs are easier to operate by primary level students (教育局，2015).

It shows that the trend of curriculum integration of coding in both European countries and Hong Kong tends to aim younger level of students. Some of the countries even treat coding as compulsory in the primary education to train students’ Computational Skills. Hong Kong and most of countries in Europe agreed that coding is only for Computational Thinking training purpose but not equipping the students programming capabilities, hence using syntax-free programs is optimal for students to purely learn Computational Thinking.

Hidden Issues of Implementing Coding to Curriculum in Hong Kong

Some organizations (智經研究中心，2015; 香港教育專業人員協會，2014) worry that Education Bureau force to implement coding into curriculum will arouse the hidden problems of increasing burden of frontline teachers and students, which will become another negative image as similar as Territory-wide System Assessment (TSA). Moreover, fixed class hours is also a major concern for teachers to balance for the new inserted coding contents. They also stated that coding should not be compulsory for primary students since only available for students who have higher capability.

Although there has no evidence shows that the coding implementation in Hong Kong aroused issues of increasing students burden, it is still a possible worry by the organizations since there are several papers (霍秉坤，2015; 陳健生、甘國臻和霍秉坤，2008; 楊思賢，2013) agree that TSA has increased the burden of both students and frontline teachers. Thus, the renewal of curriculum in Computational Thinking should be focus on junior secondary, and keep the training program as simple as possible to maintain the focus on Computational Thinking training, instead of programming ability.

Research Questions

In response to the literature review, junior secondary students are most suitable to study for the effectiveness of coding in enhancing their Computational Thinking. The training tool should be also image-based-like programs to prevent the students distracted by the syntax and let them focus on training Computational Thinking. Thus, Hour of Code is proposed as a possible tool in this study.

Hour of Code is a website to provide “one-hour introduction to computer science” (Code.org, 2016). Comparing with other coding online tutorials (codecademy, CodeCombat, etc.), Hour of Code provides game-like and syntax-free tutorials which focus on delivering techniques on Computational Thinking. Each of the tutorials is split into several sections and each tutorial is designed to an-hour length. Thus, it is flexible for teachers to implement the tutorials to their classes. Moreover, Hour of Code is one of the few organizations which set up supports in Hong Kong, schools are able to contact Hour of Code (HK) for further workshops.

With Hour of Code, this study is going to investigate the effectiveness of Hour of Code, the relationships between coding and Computational Thinking and the difficulty of learning coding skills for Hong Kong junior secondary students. Hence, the following research questions have been developed:

1. What coding skills are able to enhance computational thinking of Hong Kong junior secondary students?
2. Is Hour of Code able to deliver the coding skills to achieve the enhancement?
3. To what extend computational thinking of the students able to enhance by Hour of Code?
4. Which coding skill is the most difficult for students?

Research Methods

Research Design

To study the questions, quantitative research method would be used for the first three questions and qualitative research method would be used for the forth question. There will be a pre-test and pro-test for students before and after using Hour of Code to learning coding. The research will be held during my second teaching practice, which is October 2016 – December 2016, and the aim of research will be 1-2 form(s) of students (assume there are 30 students in a class, and depends on the class size and form size) junior secondary students, which is maximum 120 junior secondary students.

Each question in the test will correspond to one definition among the 11 definitions of Computational Thinking mentioned before in this paper. The test will include both open ended and multiple choices questions. Part of questions may not have only one correct answer, and this stimulates students to seek on more possibility to a question. Open ended questions will be answered in text which able to have a more in-depth investigation about students' thinking or understanding what method to the solutions had students used; where multiple choices questions able to evaluate any misconception of students effectively. By comparing the test results, it is able to investigate which concept(s) in Computational Thinking can be enhanced by Hour of Code.

Participants will be divided into two groups, one is the Control Group and one is the Experimental

Group. Both groups of participants should accept a pre-test (30 minutes) at the start of the research. After pre-test, two groups will have normal lessons, which the topic is not related to Computational Thinking. Instead, Experimental Group participants will be invited to use Hour of Code for two lessons (30 minutes per each) after the normal lessons, and guiding them to finish half sections of the tutorial each lesson (depends on class time). It is to balance the time limitation between normal class operation and the research. Accelerated Course – Lesson 2: Maze will be used as the teaching material, which covers with different loop and conditional statements (e.g.: repeat-until, if-then, while-do, etc.) and expected to learn Algorithm, Decomposition/Pattern Recognition and Automation among 11 Computational Thinking concepts (Code.org, 2016).

Both Control Group and Experimental Group participants will then be invited to finish a pro-test (30 minutes) as the follow up analysis, and 5-10% of the Experimental Group participants will be invited randomly to conduct a personal interview for the last research question. Interview is able to study how they feel before and after the learning, such as what skill(s) they learn from Hour of Code is/are useful in the test and finding the details of difficulty on learning the conditional statements.

Ethics Measures

Students' identity should always stay in anonymous. Thus, the study will strictly follow the guidelines provided by Human Research Ethics Committee (2014). The study will first request the consent from the principals, students and their parents. Both questionnaires and interviews will not contain any personal information of students, including name and contact number. Hashed string will be provided to students to identity the uniqueness of the collected data (hashing class and class number with SHA-1 to get the first 6 characters), such as using 356a192 to represent the first participant, da4b92 to represent the second participant, etc. All the collected data will be stored in secured devices and will be destroyed when graduate.

Possible Issues in Data Collection

There are several issues which would happen during or after the data collection. First, the language proficiency of the students is not the same, students might be distracted or getting confused by any proper noun or difficult words. Both tests and interviews would prevent using any kind of difficult words and double negatives to avoid the problem. Second, the stableness of student performance in pre-test and pro-test. Assume some students get sick during the pro-test, their performance might be decreased and the result analysis might be affected by these abnormal data. Moreover, how students learn in Hour of Code would be affected by the description and guidance provided by the teacher, the students would have worse results in post-test if they cannot have an effective learning. And last, it is possible to have data entry error when I input the data into computer for further analysis.

Data Collection

The research has been conducted at a Band 3 CMI Secondary School in Hong Kong. All of participants came from Form 2, and the age of participants distribute to age group between 13-14 and 15-17. It is known that the academic results of Control Group participants are better than the Experimental Group's from the school's supporting teachers.

At the end of data collection, 63 pairs of data have been received from participants. Control Group contributed 21 pairs of the data, and the remaining 42 pairs of the data are from Experimental Group.

Findings and Analysis

Quantitative Research

Table 1: Group Statistics (Control & Experimental)

	Group	N	Mean	Std. Deviation	Std. Error Mean
Pre-Test	Control	21	5.667	1.7416	.3801
	Experiment	42	4.233	2.2549	.3479
Post-Test	Control	21	5.714	2.0284	.4426
	Experiment	42	5.079	1.7229	.2659

According to Table 1, mean of 5.667 scores in the pre-test from Control Group, and has a net difference of 1.434 scores comparing to Experimental Group. It proved the statement from supporting teachers that claiming Control Group has better academic results than Experimental Group. In the post-test, the net difference between Control Group and Experimental Group has been reduced to 0.635 scores. The mean scores reflect Experimental Group has a well improvement on test when Control Group has no performance drop at the post-test. Apart from the mean scores, the standard deviation of Experimental Group has been reduced to 1.7229, since Control Group's has increased to 2.0284. It shows that Experimental Group has less score difference among the participants after having lessons on Hour of Code.

Table 2: Independent Samples Test (Control & Experimental)

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Pre-Test	Equal variances assumed	1.809	.184	2.553	61	.013	1.4333	.5614	.3108	2.5559
	Equal variances not assumed			2.782	50.329	.008	1.4333	.5153	.3985	2.4681
Post-Test	Equal variances assumed	.926	.340	1.301	61	.198	.6357	.4887	-.3416	1.6130
	Equal variances not assumed			1.231	34.823	.226	.6357	.5163	-.4127	1.6841

According the Independent-Samples Test (Table 2), Levene's Test of pre-test shows F = 1.809, p = 0.184 > 0.05; equal variances assumed. There was a significant score difference between Control Group and Experimental Group; t(61) = 2.553, p = 0.013 < 0.05. The t-test echoes with Table 1 that Control Group has much better results than Experimental Group in pre-test. Levene's Test of post-test shows F = 0.926, p = 0.340 > 0.05; equal variances assumed. There has no significant score difference between Control Group and Experimental Group; t(61) = 1.301, p = 0.198 > 0.05. The t-test echoes with Table 1 that the score difference between Control Group and Experimental Group has narrowed, which means Experimental Group improved their Computational Thinking concepts while the test results of Control Group have not go backwards.

Table 3: Group Statistics (Gender)

	Gender	N	Mean	Std. Deviation	Std. Error Mean
Pre-Test	M	22	4.418	2.5139	.5360
	F	20	4.030	1.9759	.4418
Post-Test	M	22	4.823	1.9400	.4136
	F	20	5.360	1.4446	.3230

Table 4: Independent Samples Test (Gender)

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Pre-Test	Equal variances assumed	2.769	.104	.552	40	.584	.3882	.7027	-1.0319	1.8083
	Equal variances not assumed			.559	39.221	.579	.3882	.6946	-1.0165	1.7929
Post-Test	Equal variances assumed	.200	.657	-1.010	40	.319	-.5373	.5322	-1.6129	.5383
	Equal variances not assumed			-1.024	38.570	.312	-.5373	.5248	-1.5992	.5246

There has also an analysis to the score difference by gender. According to Table 3, the male participants score a mean of 4.418 and female participants score a mean of 4.030. In fact, from the statistics by Table 4, Levene's Test of pre-test shows F = 2.769, p = 0.104 > 0.05; equal variances assumed. There has no significant score difference between male and female participants in pre-test; t(40) = 0.552, p = 0.584 > 0.05. In the post-test, Levene's Test shows F = 0.200, p = 0.657 > 0.05; equal variances assumed, and male participants obtain a mean of 4.8923 scores, where female participants score 5.360. There has still no significant score difference between male and female participants in the post-test; $t(40) = -1.010$, $p = 0.319 > 0.05$. The statistics show male and female participants perform similarly in Computational Thinking.

Table 5: Paired Samples Correlations

		N	Correlation	Sig.
Control	Pre-Test & Post-Test	21	.651	.001
Experimental	Pre-Test & Post-Test	42	.451	.003

Table 6: Paired Samples Tests

		Paired Differences							
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed)
Control	Pre-Test - Post-Test	-.048	1.596	.348	-.774	.679	-.137	20	.893
Experimental	Pre-Test - Post-Test	-.8452	2.1327	.3291	-1.5098	-.1806	-2.568	41	.014

In the Paired Samples Test (Table 6), it is able to observe that there has no significant change between pre-test and post-test in Control Group; $t(20) = -0.137$, $p = 0.893 > 0.05$. However, Experimental Group got a significant change in the test results between pre-test and post-test; $t(41) = -2.568$, $p = 0.014 < 0.05$. The statistics prove Hour of Code is effective in enhancing participants' Computational Thinking concepts again.

Since it is able to see the effectiveness of Hour of Code in enhancing participants' Computation Thinking concepts, there has a test result on each question with comparing Control Group and Experimental Group. Question 9 and Question 13 of the test will be discussed later in this article.

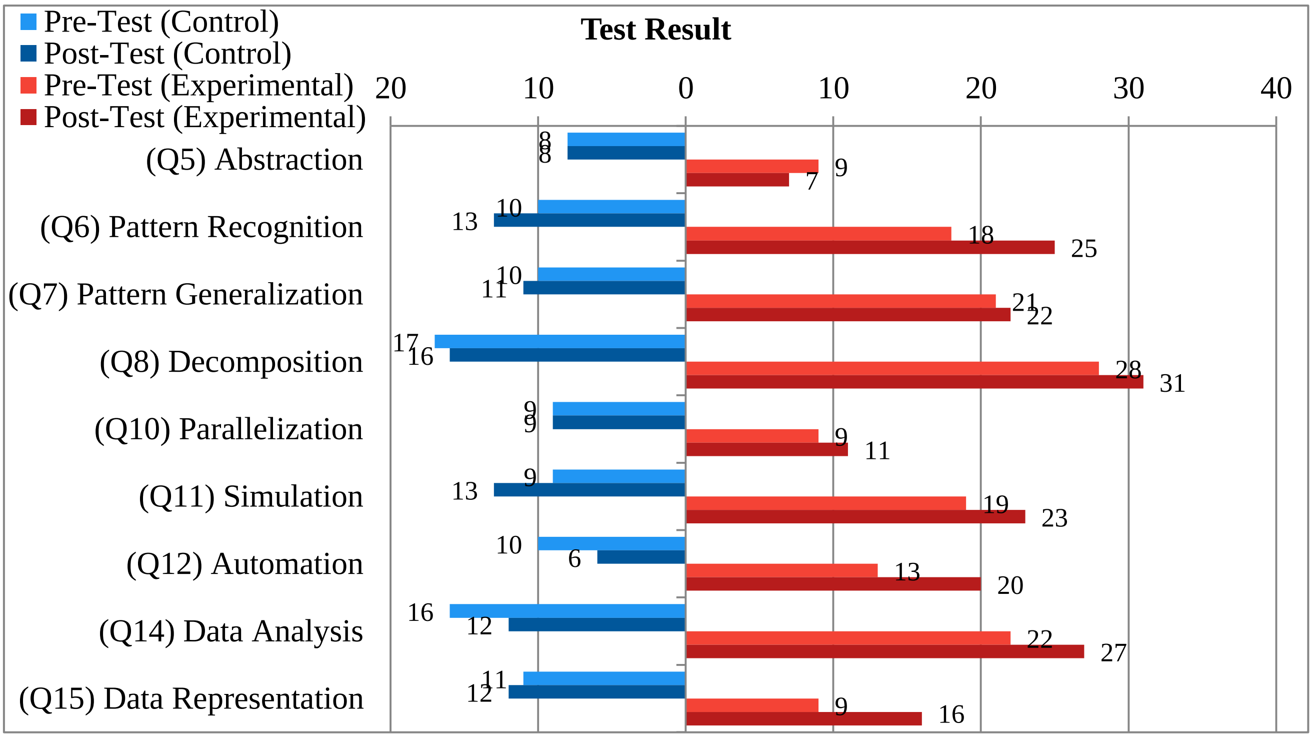
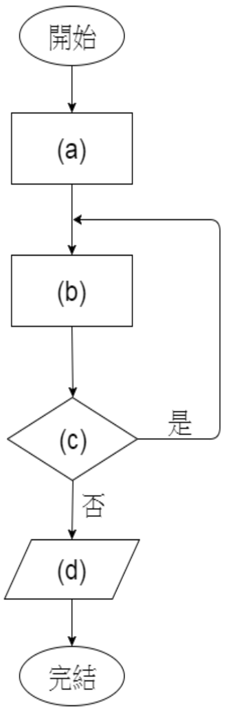


Fig. 1: Test Result Comparison of Each Test Question by Control and Experimental Group

In the question where examine participants Abstraction, Control Group has 8 participants answered the questions correctly in the pre-test, and 8 in the post-test (+0); Experimental Group has 9 participants correctly answered the question in pre-test, and 7 in the post-test (-2). In question 6 (Pattern Recognition), Control Group has increased 3 correct answers from 10 correct answers in the pre-test to 13 correct answers in the post-test; Experimental Group has increased 7 correct answers from 18 correct answers in the pre-test to 25 correct answers in the post-test. In question 7 (Pattern Generalization), Control Group and Experimental Group has both increased 1 correct answer, where Control Group from 10 correct answers in the pre-test to 11 correct answers in the post-test, and Experimental Group from 21 correct answers in the pre-test to 22 correct answers in the post-test. In question 8 (Decomposition), Control Group has decreased 1 correct answer from 17 answers to 16 answers; Experimental Group has increased 3 correct answers from 28 answers to 31 answers. In question 10 (Parallelization), there has no change of correct answer in Control Group, which has 9 answers; Experimental Group increased 2 correct answers from 9 answers to 11 answers. In question

11 (Simulation), both Control Group and Experimental Group has increased 4 correct answers, where Control Group is from 9 answers to 13 answers, and Experimental Group is from 19 answers to 23 answers. In question 12 (Automation), Control Group has dropped 4 correct answers from 10 to 6; Experimental Group has increased 7 correct answers from 13 answers to 20 answers. In question 14 (Data Analysis), Control Group has dropped 4 correct answers from 16 to 12; Experimental Group has increased 5 correct answers from 22 answers to 27 answers. In question 15 (Data Representation), Control Group has increased 1 correct answer from 11 answers to 12 answers; Experimental Group has increased 7 correct answers from 9 to 16 answers.



若要計算從 1 加至 100 (1+2+3+⋯+100) 的總和，
請完成下列流程圖。

1) 指數 i 大於或等於 0 ($i \geq 0$)
2) 設總數 sum 為 0 ($sum = 0$)，指數 i 為 1 ($i = 1$)
3) 指數 i 少於或等於 100 ($i \leq 100$)
4) 設總數為 0 ($sum = 0$)，指數 i 為 1 ($i = 100$)
5) 總數 sum 加指數 i ($sum = sum + i$)，指數 i 加 1 ($i = i + 1$)
6) 總數 sum 加指數 i ($sum = sum + i$)，指數 i 減 1 ($i = i - 1$)
7) 輸出總數 sum

Fig. 2: Question 9 – Algorithm Test

In question 9, none of the participants able to correct answer the full sequence. Despite, it is able to analyze the change of answer by students according each section in the question.

According to Fig. 2, the flow chart has part (a), (b), (c) and (d) to let participants fill in the provided instruction 1-7 for computing the sum from 1 to 100, where each part should have one instruction. Part (a) is expected to fill in “input” instruction, which is the instruction 2 or 4; part (b) is expected to fill in “process” instruction, which is the instruction 5 or 6; part (c) is expected to fill in “condition” instruction, which is the instruction 3 or 1; part (d) should fill in “output” instruction, which is the instruction 7. Hence, if participants start the flow with instruction 2, the complete sequence should be 2 → 5 → 3 → 7; if participants start the flow with instruction 4, the complete sequence should be 4 → 6 → 1 → 7. To measure participants to the understanding of Algorithm, it is able to count the correct answers by the participants to each part in the flow chart.

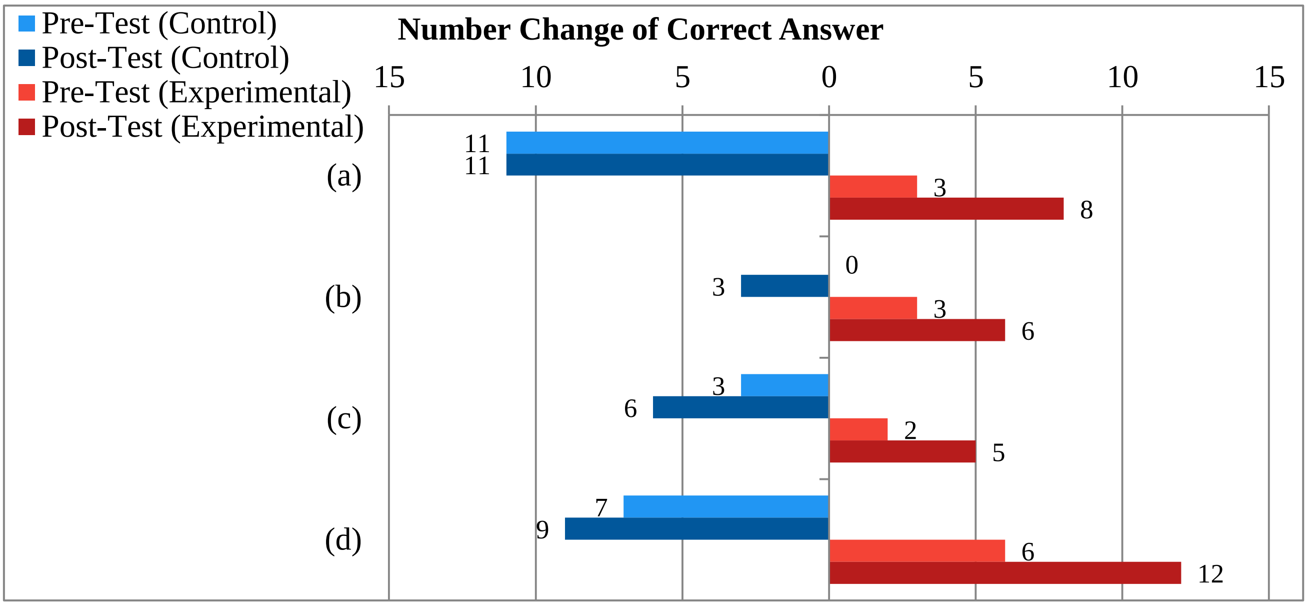


Fig. 3: Number Change of Correct Answer in Question 9

According to Fig. 3, there has 11 participants in Control Group answered 2 or 4 in pre-test for part (a), where there has no change in the post-test for Control Group; In Experimental Group, 3 participants answered part (a) in the pre-test correctly, and having an increasing 5 correct answers in the post-test to 8 correct answers. In part (b), Control Group has an increase of 3 correct answers from none in the pre-test to 3 answers in the post-test; Experimental Group has also increased 3 correct answers from 3 answers in the pre-test to 6 answers in the post-test. In part (c), both Control Group and Experimental Group has increased 3 correct answers, where Control Group is from 3 answers to 6 answers, and Experimental Group is from 2 answers to 5 answers. In part (d), Control Group has increased 2 correct answers from 7 answers to 9 answers; Experimental Group has increased 6 correct answers from 6 answers to 12 answers. From the above results, it is able to see that Experimental Group has better performance increase in the post-test, especially for part (a) and (d), which means Experimental Group enhanced the concepts of "input" and "output" after the lessons of Hour of Code.

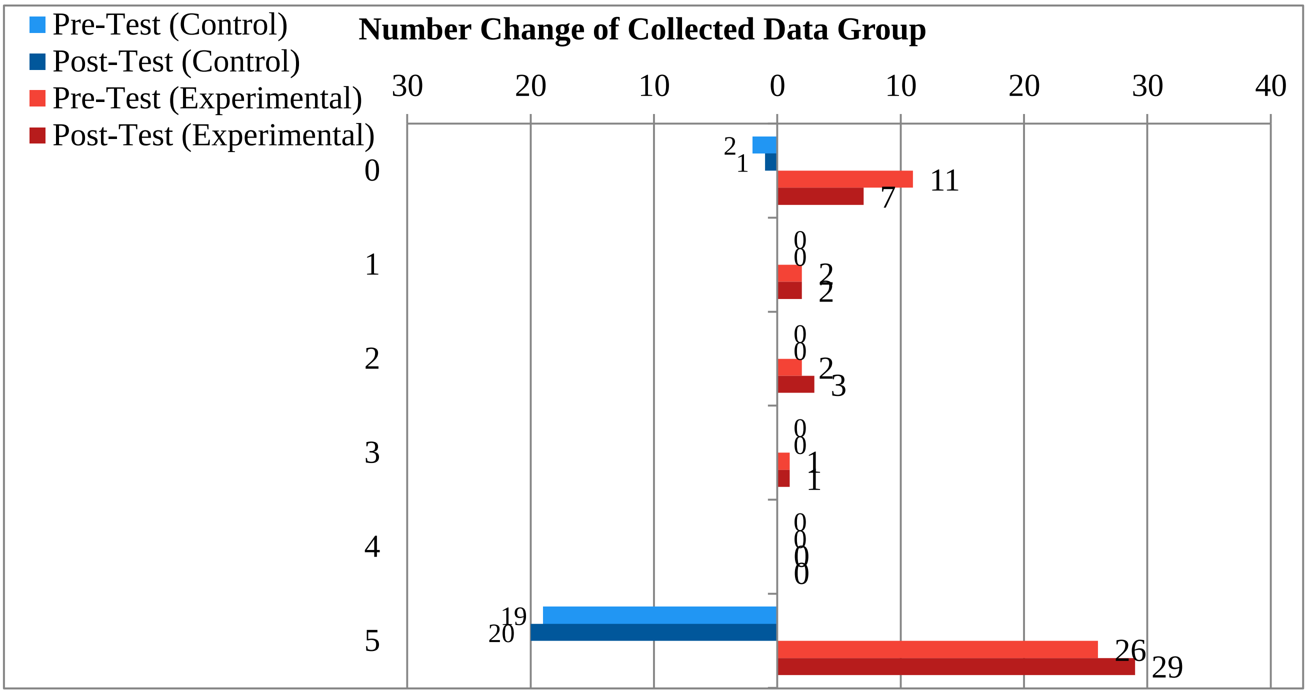


Fig. 4: Number Change of Collected Data Group in Question 13

For question 13, each group of data contains a data of person height and his/her shoes size. Any blank in the group would consider as incomplete answer and not counted into collected data group(s). According to Fig. 4, 19 participants in Control Group collected whole 5 data groups during pre-test, and added 1 participant to have 20 participants collected 5 data groups in the post-test; Experimental Group has an increase of 3 participants who collected 5 data groups from 26 participants in pre-test to 29 participants in post-test. There were 2 participants in Control Group and 11 participants in Experimental Group cannot collect any data group, but the number has deduced to 1 participant and 7 participants in Control Group and Experimental Group respectively. Control Group has no participants who collect 1 to 4 pair(s) of data group, but Experimental Group does. Experimental Group has no numerical change in collecting 1, 3 and 4 data group(s), where having 2 participants, 1 participant and no participants collected the data group respectively. In fact, there has 1 participant of increase for collected 2 data groups in Experimental Group, where from 2 participants in pre-test, to 3 participants in post-test. In general, Fig. 4 shows that Experimental Group collects more data groups can benefits to their judgements on answering question 14 and 15 in the post-test, which can explain the enhancement of test results of Experimental Group in question 14 and 15.

Qualitative Research

After the interview, the interviewees have reported the difficulties, which distribute in loop and condition functions. Most of the interviewees start feeling difficult from Stage 9 in the course, which is introducing the concept of Repeat…Until ("轉彎果個位(殭屍)", "重覆果啲難啲，第九關開始"). The interviewees tackle with the problems by thinking step by step ("試驗，一個一個試"), to firstly find out a complete sequence of doing the task, then generalizing the sequences into patterns. Apart from the difficulties of the contents, all of the interviewees though Hour of Code is interesting and attractive for them to continue the lessons ("我覺得好得意", "有興趣玩"). The game-like contents in Hour of Code motivate students to learn the lessons in self-driven, and this has largely assisted the teachers in classroom management since more students would focus on the course. Part of the interviewees also feels the course could help their learning in Mathematics because of better reaction in thinking ("轉數快咗"). However, they cannot link up the knowledge into daily life.

Limitations and Suggestions

The research has several limitations which affects the preciseness of the data analysis.

1. **Question design effects corresponding result**

   Some interviewees reflect they barely understand question 9. Question 9 and other questions may affect the ability of examination to participants Computational Thinking due to the question design. More detailed describe to each question and further studying on question design is recommended.

2. **Insufficient questions for measurement**

   Since there has only 1 question for measuring each Computational Thinking concepts, any errors in question design may impact the test results. More questions would prove higher fault tolerance in question design. It is recommended to provide more questions for each Computational Thinking concept, depends on the time availability of the test.

3. **Unequal sample size of control and experimental group**

   In this research, it is unable to collect an equal size of samples for Control and Experimental Group. The unequal sample size might affect the preciseness of data analysis. It is recommended to balance the sample size between control and experimental group as much as possible.

4. **Small sample size may result less accurate statistics**

   Alongside with unequal sample size of Control and Experimental Group, the sample size of each group is also smaller than expected. It is recommended to get a larger sample size, ideally 60 participants for Control Group and 60 participants fir Experimental Group, to combine the reality of a school environment and testers you can get.

5. **Slightly insufficient time for test**

   According to the observation, students can barely finish the test in 30 minutes. It might not so enough for a junior secondary student. It is recommended to provide more time and let the participants have sufficient time on understanding and answering the questions.

6. **Teaching Method might affect the results**

   According to the interview, participants generally feel difficult in the topics like While…Do, Repeat…Until and If…Then these loop and condition functions. Although the time is limited for teaching all of contents, it is still recommended to spend more time on teaching specific topics (While…Do, Repeat…Until, If…Then) or try to extend the lesson time to make more coverage on those topics.

Conclusion

To sum up, Hour of Code does able to enhance the Hong Kong junior students' Computational Thinking concepts in this research. According to the analysis, Hour of Code able to enhance Pattern recognition and Automation among the Computational Thinking concepts, which is in line with the learning objectives of the selected course in general. Although there has no correct answers on Algorithm question, it is still able to see the improvement on "input" and "output" part in the analysis.

Summarizing the interviews, loop and condition functions are difficult to students, but also helpful on enhancing students' Computational Thinking concepts at the same time. Hour of Code is also attractive, which able to motivate students on learning the concepts in an easier way.

References

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology, 38*(6), 20-23.

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

Balanskat, A. & Engelhardt, K. (2015). *Computing out future: Computer programming and coding - Priorities, school curricula and initiatives across Europe (Update 2015)*. Belgium: European Schoolnet.

Chea, T. (2014, September). Facebook CEO wants more students to explore tech. *Associated Press*. Retrieved from http://news.yahoo.com/facebook-ceo-wants-more-students-223235565.html

Code.org (2013, December 8). President Obama asks America to learn computer science [Video file]. Retrieved from https://youtu.be/6XvmhE1J9PY

Code.org (2016). *Hour of Code*. Retrieved from https://code.org/learn

Code.org (2016). *Instructor Handbook of Unplugged and Online Lesson Plans: Computer Science Fundamentals for Courses 1 – 4*. Retrieved from https://code.org/curriculum/docs/k-5/complete.pdf

Coulter, B., Lee, I. & Martin, F. (2010). *Computational Thinking for Youth*. Newton, MA: Education Development Center.

Curriculum Development Council (2015). *Promotion of STEM Education - Unleashing Potential in Innovation*. Hong Kong, China: Curriculum Development Council.

CSTA & ISTE (2011). *Computational Thinking teaching resources, Second Edition*. Retrieved from https://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf

Education Bureau (2015). *Report on the Fourth Strategy on Information Technology in Education: Realising IT Potential, Unleashing Learning Power, Holistic Approach*. Hong Kong, China: Education Bureau.

Furber, S. (2012). *Shut down or restart? The way for computing in UK schools*. London, UK: The Royal Society.

Google (2016). *Exploring Computational Thinking*. Retrieved from https://www.google.com/edu/resources/programs/exploring-computational-thinking/#!home

Hu, C. (2011, June). Computational thinking: what it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 223-227). ACM.

Human Research Ethics Committee (2014). *HKIEd's Guidelines on Ethics in Research*. Hong Kong, China: The Hong Kong Institute of Education.

IBM China/Hong Kong Limited (2013). *Consultancy Services for the Digital 21 Strategy Review for the Office of the Government Chief Information Officer Strategy Report*. Retrieved from http://www.digital21.gov.hk/eng/relatedDoc/download/IBM-ConsultancyStudyReport_eng.pdf

Lee H. L. (2015, May 4). Source Code of Sudoku solver [Facebook update]. Retrived from https://www.facebook.com/leehsienloong/photos/a.344710778924968.83425.125845680811480/905828379479869/

Stephenson, C. & Barr, V. (2011). Defining Computational Thinking for K-12. *CSTA Voice*, *7*(2), 3-4.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U (2014). *Defining Computational Thinking for Science, Technology, Engineering, and Math*. Chicago, US: Northwestern University.

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

智經研究中心(2015.11.19)：推編寫程式教育 防變 TSA 翻版，《經濟日報》，檢自 http://paper.hket.com/article/995582/%E6%8E%A8%E7%B7%A8%E5%AF%AB%E7%A8%8B%E5%BC%8F%E6%95%99%E8%82%B2%20%E9%98%B2%E8%AE%8ATSA%E7%BF%BB%E7%89%88 瀏覽日期：20160329

香港教育專業人員協會(2014)：《教協會回應「第四個資訊科技教育策略」諮詢文件意見書》，檢自 https://www.hkptu.org/473 瀏覽日期：20160329

教育局(2015)：《常識科課程內的「計算思維」及教材設計理念》，檢自 http://www.edb.gov.hk/attachment/tc/curriculum-development/cross-kla-studies/gs-primary/teacher-edu-program/induction-for-new-gs-teachers-pri-20150130/20150130-2.pdf 瀏覽日期：20160329

霍秉坤(2015)：香港全港性基本能力評估(TSA)的操練：囚徒困局的視角，《教育學報(香港)》，43(2)，頁 51-70。

陳健生、甘國臻和霍秉坤(2008)：課程的「軟」和「硬」政策實施：教師層面的理解，《課程研究》，4(1)，頁 1-21。

楊思賢(2013)：學校評鑑制度：香港經驗的反思，《課程研究》，8(1)，頁 129-157。

Appendix I – Pre-Test & Post-Test

<u>個人資料</u>

1. 研究編號

____

2. 性別

☐男 ☐女

3. 年齡

☐10-12 歲 ☐13-14 歲 ☐15-17 歲 ☐18 歲或以上

4. 就讀級別

☐中一 ☐中二 ☐中三

<u>測試題目</u>

5. 你剛買了一把新雨傘並打算向朋友介紹，你會用以下哪個範疇形容雨傘？
   (a) 顏色
   (b) 價錢
   (c) 銷售量
   (d) 長度

☐(a)、(b)及(c) ☐(a)、(b)及(d) ☐(a)、(c)及(d) ☐(b)、(c)及(d) ☐以上皆是

6. 以下為 ROT13 替換式密碼表。若要加密原始訊息「NWCSS」，加密後哪個密碼才是正確？

A	B	C	D	E	F	G	H	I	J	K	L	M
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

☐PURFF ☐NWCSS ☐PYNFF ☐AJPFF

7. 以下哪個數型符合 $n \times 2$ 的要求？
   (a) 1,3,5,7,9
   (b) 2,4,8,16,32
   (c) 2,4,6,8,10
   (d) 3,6,12,24,48

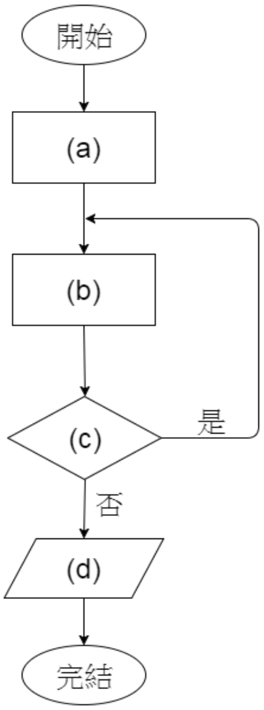
☐(a)及(b) ☐(b)及(c) ☐(b)及(d) ☐(a)及(d)

8. 你現在肚子餓打算煮即食麵吃，哪個次序是正確步驟？
   (a) 慢火煮 3-4 分鐘
   (b) 把開水煮滾
   (c) 放入麵線及調味料
   (d) 放入適量開水

☐(d)→(c)→(b)→(a) ☐(d)→(c)→(a)→(b) ☐(d)→(b)→(c)→(a) ☐(d)→(a)→(b)→(c)

Appendix I – Pre-Test & Post-Test

9. 若要計算從 1 加至 100 (1 + 2 + 3 + ⋯ + 100) 的總和，請完成下列流程圖。



1 指數 i 大於或等於 0 (i ≥ 0)
2 設總數 sum 為 0 (sum = 0)，指數 i 為 1 (i = 1)
3 指數 i 少於或等於 100 (i ≤ 100)
4 設總數為 0 (sum = 0)，指數 i 為 1 (i = 100)
5 總數 sum 加指數 i (sum = sum + i)，
指數 i 加 1 (i = i + 1)
6 總數 sum 加指數 i (sum = sum + i)，
指數 i 減 1 (i = i − 1)
7 輸出總數 sum

(a) ____________ (b) ____________ (c) ____________ (d) ____________

10. 你現在準備煮一頓晚飯，所需步驟為洗米、煮飯、洗菜、煮菜、切肉及蒸肉餅。已知洗米需時 3 分鐘、煮飯需時 15 分鐘、洗菜需時 5 分鐘、煮菜需時 7 分鐘、切肉需時 4 分鐘、蒸肉餅需時 15 分鐘。你家廚房擁有 1 個電飯煲及 2 個可供煮食的爐頭。請問這頓晚飯最快需時多久才能完成？

☐49 分鐘 ☐27 分鐘 ☐22 分鐘 ☐23 分鐘

11. 你是一間水果店的老闆。某天你的員工把 3 個果籃的標記(橙、蘋果、橙及蘋果)全部搞亂。由於果籃內的水果都包裝好不能直接從外表分析是哪種水果，只能於其中 **1 個**果籃抽取 **1 個**水果打開包裝去重新放置果籃的標記。你會選擇哪 1 個果籃抽取水果去決定 3 個果籃的標記？


橙


蘋果


橙及蘋果

以上果籃標記全部配對錯誤!

☐橙 ☐蘋果 ☐橙及蘋果

12. 你是一名圖書管理員，需要簡化工作程序處理大量讀者還書。以下哪項最能善用電腦簡化(自動化)工作程序？
    (a) 於電腦使用文本記下書本資料
    (b) 使用鍵盤輸入書本序號到電腦系統
    (c) 使用相機拍下書本封面到電腦
    (d) 使用掃描器掃描書本條碼到電腦系統

☐(a) ☐(b) ☐(c) ☐(d)

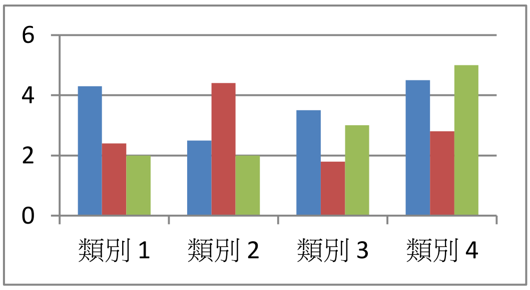
13. 請訪問你身邊的五位同學，並完成下列表格。

	身高 (cm)	**鞋子尺碼**
1		
2		
3		
4		
5		

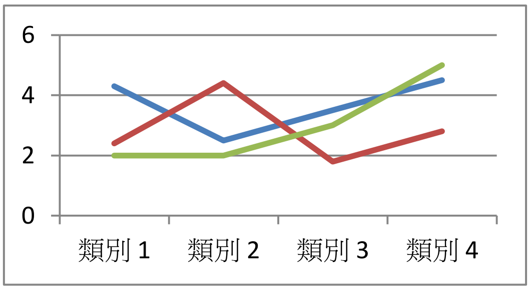
14. 承上題(13)，表格內數據有何特點？
    (a) 身高越低，鞋子尺碼越大。
    (b) 身高越低，鞋子尺碼越小。
    (c) 身高越高，鞋子尺碼越大。
    (d) 身高越高，鞋子尺碼越小。

☐(a)及(c) ☐(a)及(d) ☐(b)及(c) ☐(b)及(d)

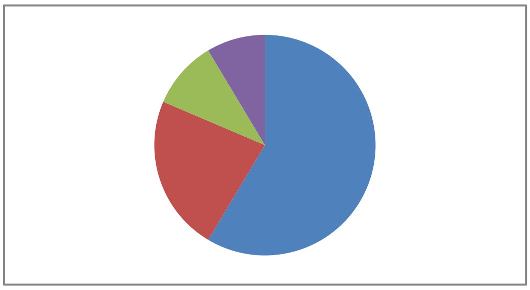
15. 承上題(14)，以下哪種圖表最能表達數據特點？

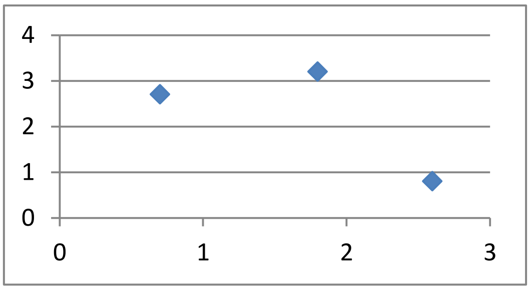


直線圖



折線圖


圓形圖



散佈圖

☐直線圖 ☐折線圖 ☐圓形圖 ☐散佈圖

Interview Question Bank

1. 你認為課程中哪部份覺得較難？

2. 你認為課程有何改善地方？

3. 你認為教師能清楚教授課程內容嗎？如否，哪部份未能清楚教授？

4. 你認為課程對日常學習有幫助嗎？如有，哪科目有幫助？

5. 你認為課程對完成測驗有幫助嗎？如有，哪部份有幫助？

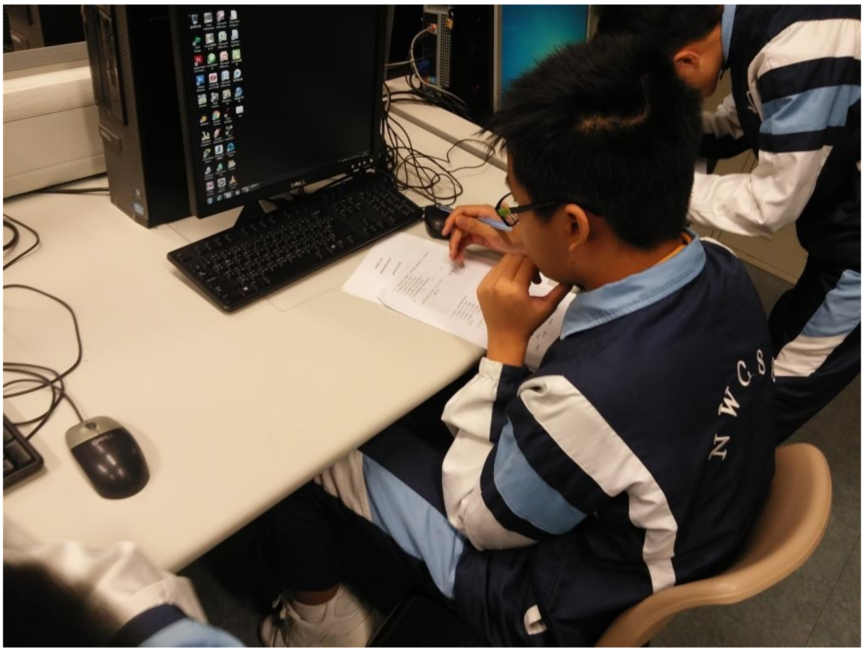
6. 你認為測驗中哪部份較難？

7. 你認為測驗有何改善地方？

8. 你對此研究有否任何疑問？

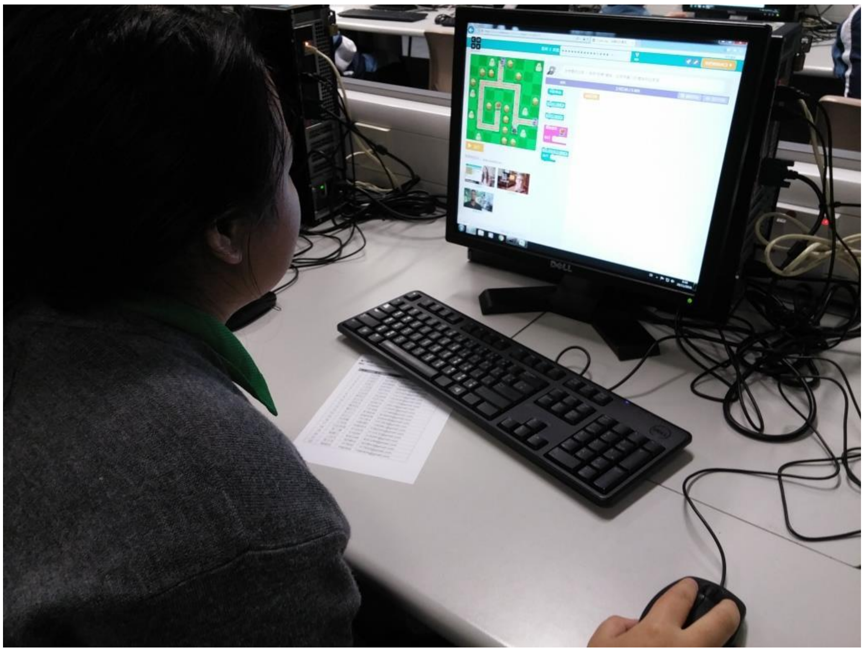
Appendix III – Research Evidence

Pre-Test & Post Test



Appendix III – Research Evidence

Hour of Code



香港教育大學

數學與資訊科技學系

參與研究同意書

探討 Hour of Code 對改進香港初中學生計算思維的成效

本人 ________________同意參加由**蘇永華博士**負責監督,**林廸**執行的研究項目。

本人理解此研究所獲得的資料可用於未來的研究和學術發表。然而本人有權保護自己的隱私，本人的個人資料將不能洩漏。

研究者已將所附資料的有關步驟向本人作了充分的解釋。本人理解可能會出現的風險。本人是自願參與這項研究。

本人理解我有權在研究過程中提出問題,並在任何時候決定退出研究，更不會因此而對研究工作產生的影響負有任何責任 。

參加者姓名: ________________________________

參加者簽名: ________________________________

日期: ________________________________

有關資料

探討 Hour of Code 對改進香港初中學生計算思維的成效

誠邀閣下參加蘇永華博士負責監督,林廸負責執行的研究計劃。她/他們是香港教育大學學生/教員。

研究計劃簡介

教育局於二零一五年開始積極推動 STEM 教育的發展,而運算思維一直被指對 STEM 教育有關鍵連繫。是次研究計劃目的為探討 Hour of Code 對改進香港初中學生計算思維的成效,以加強學生接受 STEM 教育時的解難能力。由於閣下為是次研究計劃的目標群(初中學生),故現誠邀閣下參與是次研究計劃。

研究方法

是次研究計劃對比組別期望達一百二十名初中學生,並於課堂前後確保未有阻礙正常學習情況下覓適目標群,然後進一步聯絡有意自願參與是次研究的目標群。

參加者開始參與是次研究計劃時將接受一次前測評估閣下計算思維的能力。完成前測之後參加者將參與 Hour of Code 課程,再進行後測以評估課程對改進計算思維的成效。部份(參與人數十分之一)參與者將以抽籤形式邀請進行額外小組面談,以便更深入了解參與者對是次研究的安排、成效及所遇困難。

是次研究將分開三天進行。第一天參與者會進行為時約半小時的前測;第二天參與者會參與為時約一小時的 Hour of Code 課程;第三天參與者會參與為時約半小時的後測。抽籤被抽中的參與者會額外安排日期進行約半小時的小組面談。

參與者將獲得零食作為是次研究之補償。是次研究所搜集數據將對研究有關課程為學生計算思維的改進成效提供寶貴的資料。

潛在風險

閣下的參與純屬自願性質。是次研究不會對閣下造成任何風險。閣下享有充分的權利在任何時候決定退出這項研究,更不會因此引致任何不良後果。凡有關閣下的資料將會保密,一切資料的編碼只有研究人員得悉。

Appendix IV – Sample of Consent Form (Participants)

研究結果發佈

研究結果將以論文形式向香港教育大學發佈。

如閣下想獲得更多有關這項研究的資料,請與林廸聯絡,電話 或聯絡她/他們的導師蘇永華博士,電話 。

如閣下對這項研究的操守有任何意見,可隨時與香港教育大學人類實驗對象操守委員會聯絡(電郵: ; 地址:香港教育大學研究與發展事務處)。

謝謝閣下有興趣參與這項研究。

林廸
首席研究員

Appendix V – Sample of Consent Form (Parents)

香港教育大學

數學與資訊科技學系

參與研究同意書

探討 Hour of Code 對改進香港初中學生計算思維的成效

茲同意敝子弟 ________________ 參加由**蘇永華博士**負責監督，**林廸**執行的研究項目。

本人理解此研究所獲得的資料可用於未來的研究和學術發表。然而本人有權保護敝子弟的隱私,其個人資料將不能洩漏。

研究者已將所附資料的有關步驟向本人作了充分的解釋。本人理解可能會出現的風險。本人是自願讓敝子弟參與這項研究。

本人理解本人及敝子弟皆有權在研究過程中提出問題,並在任何時候決定退出研究，更不會因此而對研究工作產生的影響負有任何責任。

參加者姓名: ________________

參加者簽名: ________________

父母姓名或監護人姓名: ________________

父母或監護人簽名: ________________

日期: ________________

Appendix V – Sample of Consent Form (Parents)

有關資料

探討 Hour of Code 對改進香港初中學生計算思維的成效

誠邀閣下及 貴子女參加蘇永華博士負責監督，林廸負責執行的研究計劃。她/他們是香港教育大學學生/教員。

研究計劃簡介

教育局於二零一五年開始積極推動 STEM 教育的發展，而運算思維一直被指對 STEM 教育有關鍵連繫。是次研究計劃目的為探討 Hour of Code 對改進香港初中學生計算思維的成效，以加強學生接受 STEM 教育時的解難能力。由於 貴子女為是次研究計劃的目標群(初中學生)，故現誠邀閣下及 貴子女參與是次研究計劃。

研究方法

是次研究計劃對比組別期望達一百二十名初中學生，並於課堂前後確保未有阻礙正常學習情況下覓適目標群，然後進一步聯絡有意自願參與是次研究的目標群。

參加者開始參與是次研究計劃時將接受一次前測評估閣下計算思維的能力。完成前測之後參加者將參與 Hour of Code 課程，再進行後測以評估課程對改進計算思維的成效。部份(參與人數十分之一)參與者將以抽籤形式邀請進行額外小組面談，以便更深入了解參與者對是次研究的安排、成效及所遇困難。

是次研究將分開三天進行。第一天參與者會進行為時約半小時的前測；第二天參與者會參與為時約一小時的 Hour of Code 課程；第三天參與者會參與為時約半小時的後測。抽籤被抽中的參與者會額外安排日期進行約半小時的小組面談。

參與者將獲得零食作為是次研究之補償。是次研究所搜集數據將對研究有關課程為學生計算思維的改進成效提供寶貴的資料。

潛在風險

閣下及 貴子女的參與純屬自願性質。是次研究不會對閣下及 貴子女造成任何風險。閣下及 貴子女享有充分的權利在任何時候決定退出這項研究，更不會因此引致任何不良後果。凡有關 貴子女的資料將會保密，一切資料的編碼只有研究人員得悉。

Appendix V – Sample of Consent Form (Parents)

研究結果發佈

研究結果將以論文形式向香港教育大學發佈。

如閣下想獲得更多有關這項研究的資料,請與林廸聯絡,電話　　　　或聯絡她/他們的導師蘇永華博士,電話　　　　。

如閣下或　貴子女對這項研究的操守有任何意見,可隨時與香港教育大學人類實驗對象操守委員會聯絡(電郵:　　　　　; 地址:香港教育大學研究與發展事務處)。

謝謝閣下有興趣參與這項研究。

林廸
首席研究員

Appendix VI – Sample of Consent Form (Schools)

香港教育大學

數學與資訊科技學系

參與研究同意書(學校)

探討Hour of Code對改進香港初中學生計算思維的成效

本校同意參加由**蘇永華博士**負責監督,**林廸**負責執行的研究計劃。她/他們是香港教育大學學生/教員。

本人理解此研究所獲得的資料可用於未來的研究和學術發表。然而本人有權保護本校學生/教師的隱私,其個人資料將不能洩漏。

研究者已將所附資料的有關步驟向本人作了充分的解釋。本人理解可能會出現的風險。本人是自願讓本校學生/教師參與這項研究。

本人理解本人及本校學生/教師皆有權在研究過程中提出問題,並在任何時候決定退出研究，更不會因此而對研究工作產生的影響負有任何責任。

簽署: __________________________

校長/ 學校代表*姓名: __________________________ (教授/博士/先生/女士/小姐*)

職位: __________________________

學校名稱: __________________________

日期: __________________________

(**請刪去不適用者*)

Appendix VI – Sample of Consent Form (Schools)

香港教育大學
數學與資訊科技學系

參與研究同意書(學校)

探討Hour of Code對改進香港初中學生計算思維的成效

誠邀 貴校參加**蘇永華博士**負責監督,**林廸**負責執行的研究計劃。她/他們是香港教育大學學生/教員。

研究計劃簡介

教育局於二零一五年開始積極推動 STEM 教育的發展，而運算思維一直被指對 STEM 教育有關鍵連繫。是次研究計劃目的為探討 Hour of Code 對改進香港初中學生計算思維的成效，以加強學生接受 STEM 教育時的解難能力。由於 貴校學生為是次研究計劃的目標群(初中學生)，故現誠邀 貴校參與是次研究計劃。

研究方法

是次研究計劃對比組別期望達一百二十名初中學生，並於課堂前後確保未有阻礙正常學習情況下覓適目標群，然後進一步聯絡有意自願參與是次研究的目標群。

參加者開始參與是次研究計劃時將接受一次前測評估閣下計算思維的能力。完成前測之後參加者將參與 Hour of Code 課程，再進行後測以評估課程對改進計算思維的成效。部份(參與人數十分之一)參與者將以抽籤形式邀請進行額外小組面談，以便更深入了解參與者對是次研究的安排、成效及所遇困難。

是次研究將分開三天進行。第一天參與者會進行為時約半小時的前測；第二天參與者會參與為時約一小時的 Hour of Code 課程；第三天參與者會參與為時約半小時的後測。抽籤被抽中的參與者會額外安排日期進行約半小時的小組面談。

參與者將獲得零食作為是次研究之補償。是次研究所搜集數據將對研究有關課程為學生計算思維的改進成效提供寶貴的資料。

潛在風險

貴校學生/教師的參與純屬自願性質。是次研究不會對貴校學生/教師造成任何風險。所有參加者皆享有充分的權利在研究開始前或後決定退出這項研究，更不會因此引致任何不良後果。凡有關 貴校學生/教師的資料將會保密，一切資料的編碼只有研究人員得悉。

Appendix VI – Sample of Consent Form (Schools)

研究結果發佈

研究結果將以論文形式向香港教育大學發佈。

如閣下想獲得更多有關這項研究的資料，請電郵與本人或本人的導師蘇永華博士 聯絡。

如閣下對這項研究的操守有任何意見，可隨時與香港教育大學人類實驗對象操守委員會聯絡(電郵： ；地址：香港教育大學研究與發展事務處)。

謝謝閣下有興趣參與這項研究。

林迪

首席研究員