Educational Effects of SW Coding Notes on Computational Thinking

Seung Ryol Maeng

***Abstract:** Computational thinking is a fundamental skill to be used by everyone in the 4th industrial revolution age and it is a hot issue of the education field in the world. Usually, SW coding means the whole process to make a computer program. Since a program has logical structure and its development process requires subsidiary abilities which analyze a problem and devise the stepwise procedure, SW coding has been understood to be a good method for computational thinking(CT). In this paper, we propose a way to teach CT by using coding notes and analyze its educational effects on CT.*

To find the educational effects of SW coding notes, logical reasoning ability before and after SW coding education was measured through questionnaire survey and compared by statistical methods. We sampled 51 students and divided them two groups; one for computer science major(38) and the other for computer science non-major(13). They are asked to answer three categories of questions such as problem understanding, data analysis, and problem solving. An d then by the paired t-test along with normality and homoscedasticity of two groups, the change of student's CT ability before and after taking SW coding education was statistically tested at a significant level of 95%.

According to our experiments, p-values of the paired t-test for problem understanding, data analysis, and problem solving are 0.176, 0.134, and 0.470, respectively and alternative hypotheses are accepted. It implies that student's logical reasoning ability can be improved and their academic achievement is also relatively great by using SW coding notes. Although there are similar results for educational effects of SW coding, our work is different from them in terms of using new education tool named as SW coding notes.

***Keywords :** Computational Thinking, Logical Reasoning, Data-oriented Problem Solving, Computing-oriented Problem Solving, Algorithm.*

I. INTRODUCTION

Ten years ago, professor Wing suggested the necessity of computational thinking(CT) [1]. According to her work, CT represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. As the 4th industrial revolution age is coming her suggestion is being meditated in the educational world [2].

Many experts expect computer literacy will dominate one's competitiveness in the coming 4th Industrial revolution age.

Many countries are adopting computational thinking in K-12 curriculum with a form of the SW education [3]. Korea also started the SW education at middle school in previous year and is going to expand it to primary school in this year. In these days, SW education methods such as an unplugged way, physical computing, algorithm, and visual programming are widely used in the primary and middle school. Many researchers reported that visual programming tools such as Scratch, Raptor, Entry, and Kudo have an effect on the improvement of logical abilities of students, especially primary students [4,5]. At this time, our question is "are typical ones best ways for computational thinking?" Some research results proposed that the SW education is desirable to be tied with the STEAM teaching method [6]. It means that CT must be for overall thinking, not only for logical thinking. In this viewpoint, just only one of typical ones is not good for CT.

Most of universities have not yet payed attention to computational thinking education. Introduction of CT to university curriculum must not be postponed anymore because the 4th industrial revolution has already come in our life. It motivated our work. In addition, simplification of SW education to physical computing or visual programming may not be best for CT. Particularly, these ways in the primary and middle school must be differentiated from ones of the university. Based on our work, it is expected that CT will introduce to the curriculum for the college of engineering to a minimum.

Remember Wing's suggestion in which computation is the automation of abstractions. Since automation is completed by software, software education can be a way for CT. As we know, software can be developed through a sequence of steps. We can group these steps into two categories such as data-oriented one and computing-oriented one. I want to refer to the former as data-oriented problem solving and the latter as computing oriented one. Someone may want to call the former a data modeling and the latter an algorithm. Without loss of generality, a given problem should be solved by both the data-oriented problem solving and the computing-oriented one. Therefore SW education must be targeted to enhance two abilities and the effects of SW education should be measured without loss of this point.

In the center of two problem solving abilities, we have attention to the SW development processes. Remember that the reason why we teach SW coding is that each step of the SW development has an element for CT. There is only a problem such that SW coding is not easy and needs much time to learn. To overcome this problem, we devised the coding notes, which consists of templates for overall steps, and used it to teach SW coding. In this work,

Revised Manuscript Received on July 22, 2019.

***** Correspondence Author

Seung Ryol Maeng*, Dept. of Computer Science and Engineering, Kongju National University, 1223-24 Cheonan-Daero Seobuk-Gu Cheonan-City, Chungnam 31079, South Korea. Email: srmaeng@kongju.ac.kr

we will analyze the educational effects of SW coding notes on CT. We divided CT into three components such as the awareness, the ability, and the applicability. Although all of these components are important, our work has focused on the second component, the ability of CT, such as insight of a given problem, analytical skill of data, and logical power for solving a problem. After teaching students SW coding using the coding notes for 20 hours, the change of their abilities for CT will be measured. Conclusively, it can be found that their abilities for CT were significantly improved by using SW coding notes. As a reference, a survey of the first component, awareness of Korean university students on CT, was reported in the ISER conference [7].

This paper contains three sections. Next section illustrates design of question for our experiment. In the following section, data aggregation and its analysis will be given. Finally, we will present the statistical interpretation on the analysis.

II. STRUCTURE OF SW CODING NOTES

A. SW Coding Processes

As shown in Table 1, software is made through a sequence of steps, called as SW coding. That is, SW coding means the whole development processes of software. For convenience, we will split the whole processes into two parts such as data-oriented problem solving and computing-oriented one and separately illustrate them in the center of the key point. In addition, the expected effect for each step will be given.

Let's begin from the data-oriented problem solving. We refer to four steps from the problem analysis step to the data representation step as data-oriented problem solving. In the first step, data and computation elements from a given problem are extracted and they are systematically represented, consisting of input, output, and a main task. That makes a complex problem simple. Data acquisition step is to gather the detailed data for solving the problem, based on the three elements extracted in the previous step. Data analysis step is to survey the properties of gathered data such as a name, a value range, related data, and so on. These properties will be used to decide a data type and its representation. Next step is the data representation, in which the storage types for the data are decided.

Table- I: SW coding processes and educational effects

Step	Function	Problem solving	Effects
1	Problem Analysis	Data-oriented	Insight & problem simplification
2	Data Gathering		Observation
3	Data Analysis		Analytical skill
4	Representation		Knowledge representation
5	Problem Division	Computing-oriented	Process, labor division & effectiveness
6	Abstraction		Modularity
7	Algorithm		Sequence, branch & iteration
8	Coding		Computer language
9	Testing		Simulation & Validation

From now on, we will focus on the computing-oriented problem solving. A sophisticated problem is divided into several sub-problems in the problem division step. A main task of each sub-problem is defined at the abstraction step together with input and output. In the algorithm step, computing procedures for each abstracted sub-problem, which include sequential and conditional commands, are constructed step by step. Algorithm can be represented with various ways such as pseudo-code and flow chart. In general, testing step is performed for software code. Logical errors of the code, especially the malfunctions of an algorithm, are usually caused by the whole processes of SW coding. Through repeated error correction, algorithm is correctly completed.

B. Construction of Coding Notes

Simply speaking, SW coding notes is a template in the form of a table, which of row and column specifies the contents to be done at each step [8]. Although jobs for each step are well defined, students cannot easily do the defined work without experience. For an example, the skeleton and requirements for a given problem is analyzed and restructured to be a system with input and output at the first step of SW coding, the problem analysis step. However, students cannot easily understand how to work although this explanation is simple. As we can conveniently draw the line on a given dotted line, a template for SW coding plays a role like a dotted line. Table 2 shows a template for the problem analysis step of the coding processes.

For every step of the coding processes, the corresponding template has been devised. Each template consists of three parts; the first part describes the job to be done in this step and the second one includes the main job. And the last one gives memo space for this step.

C. Educational of SW Coding using the Coding Notes

As mentioned in previous sections, CT ability can be enhanced through education of SW coding processes. Of course, explanation of each step can be done in words. In this case, although enough explanation about what must be done in each step is given, most of students feel difficulty to do that. To overcome this difficulty, we need a new teaching method and tool. Our idea for a new method was borrowed from the sketchbook for children. Little kids feel difficulty to draw a figure consisting of lines and circles. So the sketchbook for kids has usually the figures preprinted with dotted lines and kids exercise drawing on the dotted lines. Similarly when a guideline for SW coding is given, the difficulty of SW coding can be reduced.

We instruct students the usage of the coding notes in the viewpoint of the data-oriented and the computing-oriented problem solving. As a simple example, when the problem calculating the area of a rectangle is given, students write how to automatically calculate it in the coding notes. Writing leads to thinking and is a course for the final result. For our experiment, we teach the usage of the coding notes and instructed students to practice the given some problems. After sufficient time, we show students an answer sheet and make students to compare their results with the answer sheet. Finally students have discussed about the comparing results.

Table-II: An example of coding notes for the problem analysis

Description	- Extract data and computing elements from a given problem - Make a system with the extracted elements by linking input/output and a main task		
Analysis Method			
Problem element	Subclass	Contents	Etc.
	Data element		
	Computing element		
Problem definition	Input	Computation	Output
Memo			

III. EXPERIMENTAL METHOD

A. Design of Survey

The group assessment of logical thinking(GALT) is widely used as a tool to measure student's logical thinking level [4,9,10]. It includes the detailed items for evaluating the logical ability such as conservation, proposition, stochastic, correlation, and combinatorial logic. As mentioned in the previous section, CT emphasizes on the data-oriented thinking and the computing-oriented thinking while logical thinking does on the computing-oriented thinking. It can restrict our work and leads inappropriate analysis of our work for short term.

Table-III: Design of questions

Evaluation components	Measurement elements	Related educational effects	Questions
Awareness	Necessity & purpose	-	Recognition of the necessity of CT education Understanding of the objective of CT education
Problem solving	Problem understanding	Insight	Problem analysis and simplification
	Data modeling	Analytical skill	Data extraction & expression ability for problem solving
	Algorithm	Sequence & branch	Problem decomposition Logical reasoning
Applicability	Applicability	Study & living life	Application to major study Agreement of introduction CT to university curriculum

Remember that CT is not only for logical reasoning but also for integrated reasoning with STEAM. In Korea, primary and middle school teachers are instructed various coding methods for STEAM by government [11]. With this reason, SW coding is recognized to be a good way for educating CT. As summarized in Table 1, each step in the coding processes gives us various educational effects. Although it may be subjective, the key point of each SW coding process says what are performed in the corresponding step.

Our experiment is based on the academic achievement test. Table 3 shows the considerations for the test design. There are three main evaluation components such as awareness, ability, and applicability in the first column. Awareness is for assessing how students understand the significance of CT. Problem solving is for doing the student's analytical and logical capability. Applicability is included to confirm the usability of CT in student's life and study. The second column contains the measurement elements for each evaluation component. The third column includes the expected educational effects for the measurement elements. Finally, the fourth column illustrates the key point of questions for each measurement element.

In this paper, we focus on the problem solving component. So, detailed questions made for our experiment belong to three elements such as problem understanding, data modelling, and algorithm. Achievement of each question is divided into 10 levels with interval 10. Basically, test method is to write an answer on the answer sheet. To assess student's academic achievement, reference answer sheet for each question is prepared so that it contains necessary and sufficient answer item depending on the achievement level. Finally, a student's answer sheet is assessed by comparing the reference answer sheet. If a student's answer for a question includes all necessary and sufficient items, then he/she gets the academic achievement level 10 for this question. Otherwise he/she gets achievement level under 10, depending on answered items.

Table-IV: Summary of test results

No.	Test	Scores										Total	Above 50	Rate (%)
		10	20	30	40	50	60	70	80	90	100			
①	Pre	8	10	13	8	6	2	2	2	0	0	51	12	23.5
	Post	2	1	4	7	9	7	5	11	3	2		37	72.5
	Total	10	11	17	15	15	9	7	13	3	2		-	-
②	Pre	6	9	18	7	5	3	2	1	0	0	51	11	21.5
	Post	0	2	7	3	7	9	9	9	4	1		39	76.4
	Total	6	11	25	10	12	12	11	10	4	1		-	-
③	Pre	8	10	11	6	7	3	2	3	1	0	51	16	31.3
	Post	2	1	8	5	6	7	8	8	3	3		35	68.6
	Total	10	11	19	11	13	10	10	11	4	3		-	-

estions	① Problem understanding - I can extract computing factors from a problem; input, output, and main computing factor ② Data-oriented problem solving -I can derive detailed data from a problem -I can analyze properties of the detailed data: values, types, storage type and so on ③ Computing-oriented problem solving -I can divide a problem into small ones -I can express a small problem with a function type -I can make the procedure for solving the small problem: algorithm

B. Sample Data and Experimental Scope

Without loss of generality, we selected an arbitrary class, consisting of CE-major and non CE-major students, as a sample. There are 51 students in the class but their grade is not restricted to just first grade. Roughly speaking, CE-major students are 38, including 33 first grade students while non CE-major students are 13, including 5 first grade students.

As mentioned in the previous section, one of the main concerns is to test there exists the change of student's ability on CT after SW coding education. Table 4 shows the academic achievement before and after SW coding education for six questions of three measurement elements. Now we will test the change of student's CT ability for this sample.

C. Experimental Processes

For experiment, students were given an occasional training of SW coding for 20 hours. Education material for training includes the basics such as the definition of the CT, problem solving by a computer, the difference of mathematical logic and algorithmic logic, the structure of a computer and its working principles, data extraction from a problem, algorithm for problem solving, and so on.

Figure 1 shows our experiment process. In first step, the method of using SW coding notes was given and then students were tested for a simple problem in the viewpoint of data-oriented and computing-oriented problem solving. For an example, after showing two rectangles, students are asked which one is big. Then they should fill out the coding notes without any hints. They write their answer to each step of coding processes on the corresponding page of the coding notes step by step. For experiment, we have prepared 6 problems with the similar difficulty and students have exercised to solve the problems using the coding notes. 20 hours later, students were tested with the same method as pre-test. Finally, students marked their coding notes using the reference answer sheet and for accuracy of the inspection, we rechecked the results of the self-test done by students.

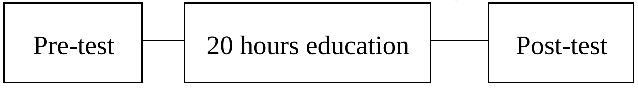

Fig. 1. Experimental Processes

IV. ANALYSIS OF EDUCATIONAL EFFECTS

A. Preparation of Analysis

As you know, significant difference between two groups can be estimated by statistical hypothesis test. Based on two assumptions such as normality and homoscedasticity, paired t-test is used for comparing the difference of the means between two paired groups. Normally, data size is more than 50 and the first requirement is satisfied. The other remains to be tested. To compare two variances of before and after groups of SW coding education, F-test was performed with the significant level 0.05. As mentioned in earlier the tests are required in the viewpoint of change of problem understanding, data-oriented, and computing-oriented solving ability. As you can see in Table 5, p-values for three test items are greater than 0.05. It fails to reject null hypotheses and two groups can be interpreted to have the homoscedasticity. Particularly, since two groups have the same size, there is no problem in using paired t-test test between two groups.

Table-V: Homogenicity of variances between two groups

Test item	df	F-value	p-value	Ratio of variance
Problem understanding	50	0.680	0.176	0.680
Data anlysis	50	0.652	0.134	0.652
Problem solving	50	0.814	0.470	0.814

B. Educational Effects of SW Coding

Intuitively, we can find from Table 4, many students have got the improvement in their CT ability through SW coding education. According to our previous work[7], above 80% of the CE-major students positively agreed CT education using the SW coding can improve their ability about data analysis and logical reasoning. Particularly, the coding notes were helpful to learn SW coding and understand CT.

Now we want to analyze the educational effects of SW coding education by comparing the means of two groups. Table 6 shows the results of paired t-test. Since all p-values are less than 0.05 and alternative hypotheses are accepted, there statistically exist the differences of the means between before and after SW coding education.

Table-VI: Results of paired t-test before and after SW coding education

Test items	df	t-value	p-value
Problem understanding	50	-9.740	3.863e-13
Data analysis	50	-9.945	1.933e-13
Problem solving	50	-8.658	1.622e-11

V. CONCLUSION

CT is one of one's capabilities required at daily life and school life in the 4th industrial revolution age. Through this experiment, we made sure that SW coding using the coding notescan be helpful to improve the problem solving capability of university students without experience of SW coding. To compare the change of their abilities before and after taking SW coding training, we carried out the paired t-test for the sample of size 51. According to the survey, there is the change of their ability at the significant level of 95%.

CT aimed at the improvement of the integrated and logical thinking ability for everyone and everywhere. In this point of view, it must be given to all university students. What's more, this subject has to be given to students of the college of engineering unrelated to students' major because they are required creative problem solving ability. Although we had an attention to usefulness of the coding notes in this research, we will compare the useful ness of our approach for SW coding education with others in next work.

ACKNOWLEDGMENT

This work was supported by the research grant of the Kongju National University in 2018.

REFERENCES

1. Jeannette M. Wing, "Computational thinking", CACM, Mar. 2006, 33-35.
2. Sung Yul Kim and Jong Yun Lee, "Development of a curriculum for the cultivating the creative gifted and talented children of informatics", J. of Korean Association of Computer Education, 17(3), 2014. 3, 25-39.
3. Jung Sook Sung and Hyeoncheol Kim, "Analysis on the international comparison of computer education in school", J. of Korean Ass. of Computer Education, 17(3), 2014.3, 25-39
4. Tae-Hun Kim and Jong-Hoon Kim, "The effects of Kodu programming learning on logical thinking and learning interest of elementary students", J. of Korean Ass. of Computer Education, 16(3), 2013. 5, 13-22.
5. Sung Yul Kim and Jong Hun Lee, "Education of algorithms using the Raptor programming education tool", J. of Korean Ass. Of Computer Education, 18(6), 2015. 11, 23-31.
6. Sugee Kim and Chul Hyun Lee, " The three-years comparative study of effects of STEAM education education programs based on physical computing", J. of Korean Ass. Of Computer Education, 19(1), 2016. 1, 11-18.
7. Seung Ryol Maeng, "Awareness of Korean university students on computational thinking", Proc. Oof ISER, International Conf. at Vienna, 2018. 8, 16-18.
8. Seung Ryol Maeng, Computational Thinking Workbook, KNU Press, 2017. 3.
9. Soon Hee Kang, Jeon Won Noh, and Jong Yoon Park, "A comparative analysis of the GALT fuul version and short version used in the science education researches", J. of Korea Science Education, 18(3), 1998. 9, 399-413.
10. Il Kyu Yoon, Jong Hye Kim, and Won Gyu Lee, "Operational definition of components of logical thinking in problem-solving process on informatics subject", J. of Korean Ass. Of Computer Education, 13(2), 2010. 3, 1-14.
11. Ministry of Education, Tutorials on SW Education for Training the Primary-Middle School Teacher, TM2016-39, 2016P

AUTHORS PROFILE

Seung Ryol Maeng is a professor of computer science and engineering at KNU. I am interested in VR/MR applications at the industry. Recently, I focus on MR in the 5G communication environment. MR technology tend to integrate various sensing data into the solution to human living. I received Ph. D. in computer science from KAIST.