J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

The Study on Direction of the Software Education☆

- focused on the freshman students of the College of Social Sciences -

Oakyoung Han[1] Jaehyoun Kim[2*]

ABSTRACT

This paper presents direction for efficient software education. Due to the impact of the Fourth Industrial Revolution, the whole world is interested in software education. However, simply teaching how to code is not software education. The thinking abilities used in coding for software implementation are even more important. Therefore, computational thinking is getting great attention. Several institutions suggest factors for computational thinking and encourage to teach in a relevant way based on the suggestion. In this study, the verification of the factors they suggested was conducted through a questionnaire. The total of 419 freshman students of the College of Social Sciences who were taking "Computational Thinking and Software Coding" class participated in the survey at the beginning and the end of the semester. We first analyzed Wing's proposal that summarized the concept of computational thinking, and reviewed the proposal of ISTE (International Society for Technology in Education) for defining computational thinking factors for coding education, also checked on the suggestion of Google for factors necessary for software coding. As a result of research analysis, this paper suggests a direction for efficient software education.

☞ keyword : Computational Thinking, ISTE, Google, Software Education, Heat map, Cluster map

1. Introduction

The 4^{th} industrial revolution has brought the fast and revolutionary changes in education. In order to secure national competitiveness, many countries around the world emphasize software education. The range of students are very broad from kindergarten to adult. Many researchers analyze research and operational examples of university-related research, education, and industry-academic cooperation to prepare for the 4^{th} industrial revolution. In addition, a variety of studies are being conducted to explore subsequent research and development tasks for software education to ensure the development of software technology [1]. This study is also a research on software education for university students especially non-computer science major.

Computational thinking is important for software education for the preparation of the 4^{th} industrial revolution. Currently, software education is positioned to educate students the ability to solve problems logically and efficiently according to computational thinking which is a problem solving technology in computer science along with educational programming languages, robot control, physical computing, web-based education and applications, hence students are able to learn computational thinking in a variety of ways [2].

Many researchers have mentioned various factors in relation to computational thinking. Professor Wing from the Carnegie-Mellon University practically organized the concept of computational thinking [3]. International Society for Technology in Education(ISTE) presented 9 steps to implement a program by adopting computational thinking. Google, one of the hottest IT companies, has proposed four factors of computational thinking for implementing software [4]. The verification of the factors that they proposed is studied in this paper to suggest the right direction for software education.

The purpose of this paper is to make a guideline how to provide efficient software education to educate students

1 University College, Sungkyunkwan University, Seoul, 03063, Korea
2 Dept. of Computer Education, Sungkyunkwan University, Seoul, 03063, Korea
* Corresponding author (jaekim@skku.edu)
[Received 18 February 2020, Reviewed 26 February 2020(R2 25 April 2020), Accepted 11 June 2020]
☆ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW)(2015-0-00914) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation)
☆ A preliminary version of this paper was presented at APIC-IST 2019.

http://dx.doi.org/10.7472/jksii.2020.21.4.69

preparing for the 4th industrial revolution.

2. Related Work

2.1 The 4th Industrial Revolution

The world's major institutions introduced IoT, big data, cloud computing, 3D printing, healthcare, mobile devices, smart machines, 3rd platform, block chain and security technologies as the future of core technology trends [5]. Furthermore, as almost all of the large industries are heading toward smart systems, a new era of information and communications technology (ICT) is becoming more visible during the past few years. Also efficacy, expectancy and acceptance of change are related to the preparation and the creation of new opportunities for the future, and the performance of the 4th industrial revolution era [6].

The basis of all this technology and direction lies in the software. It implies that software education is important to prepare for the future. Students are required to understand software coding to prepare for the future, and they have to improve the level of computational thinking abilities through software education to be a member of software society [7].

2.2 Computational Thinking

Computational thinking is a kind of analytical thinking and it is taking an approach to solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing [8]. Professor Wing listed major 10 factors as computational thinking.

- Reformulating: transform pre-existence thought
- Recursive Thinking: solve problems with repetitive extended thinking
- Data as Code: express data as code and vice versa
- Simplifying: minimize the problem via simplification
- Abstraction: extract key thoughts with an abstract approach
- Decomposition: divide a problem into small pieces
- Pre-fetching: prepare necessary things in advance
- Resource Sharing: share limited resources
- Heuristic Reasoning: solve problems by inferring what one has experienced
- Algorithmic Thinking: think procedurally to solve problems

International Society for Technology in Education(ISTE) presented nine factors of computational thinking to implement a computer program within progression [9].

- Data Collection: gather appropriate information
- Data Analysis: make a sense of data
- Data Representation: depict and organize data in appropriate way
- Problem Decomposition: break down tasks into smaller manageable parts
- Abstraction: reducing complexity to define main idea
- Algorithms and Procedures: design series of ordered steps to solve a problem
- Automation: run a program for tasks
- Simulation: run experiments using models
- Parallelism: organize resources to simultaneously

Google insists that computational thinking is not programming. Computational thinking enables you to work out exactly what to tell the computer to do. They proposed the four cornerstones of computational thinking [10].

- Decomposition: break down a complex problem into manageable parts
- Pattern Recognition: look for similarities among and within problems
- Abstraction: focus on the important information only, ignoring irrelevant detail
- Algorithms: develop a step-by-step solution to the problem

3. Methods

3.1 Questionnaire

3.1.1 Questionnaire Survey Target

For this study, a survey was conducted on 574 students at the beginning of the semester and 419 students at the end of the semester from the College of Social Sciences, freshman students of the University located in Seoul. To measure the improvement of computational thinking, the surveys were conducted twice, at the beginning of the semester and the end of the semester.

The questionnaire consisted of 51 questions. The questionnaire asked about statistical questions, computational thinking related questions, and learner background dealing with software education.

The Google questionnaire form was applied for students to participate freely, and the students voluntarily participated in the survey.

3.1.2 Applied Class

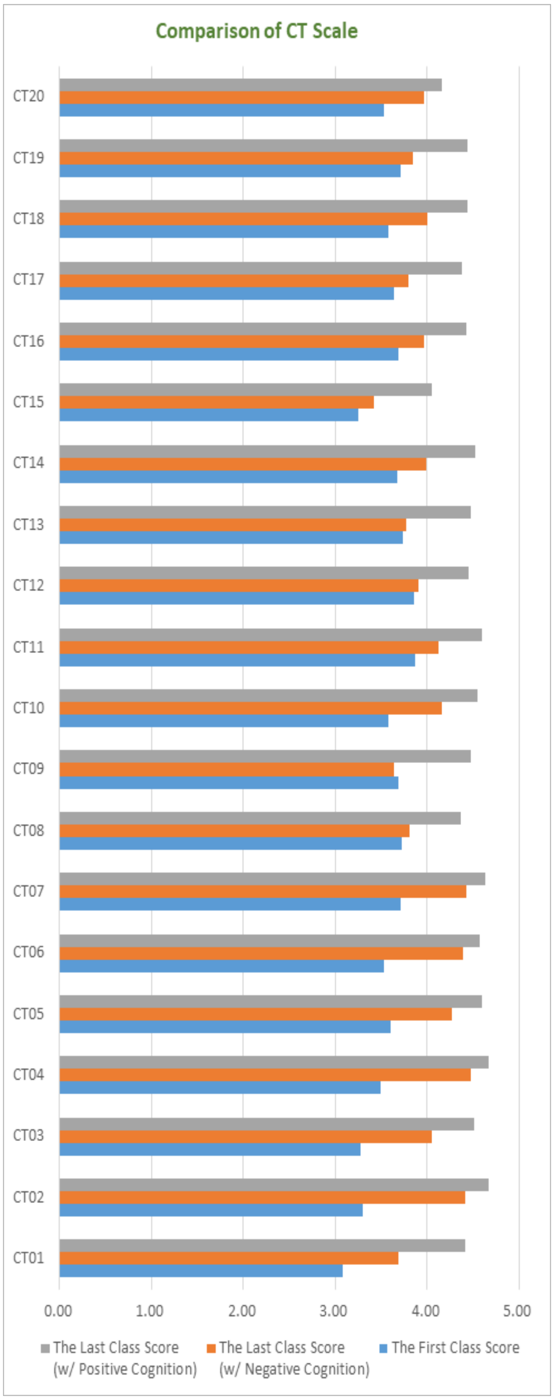
'Computational Thinking and Software Coding' is a curriculum to raise awareness of the principles of computer science, trends in the latest IT technologies and importance of information society for non-majoring students who are new to software. The main concept of computational thinking was adopted to improve logical thinking ability of the learners [11].

'Computational Thinking and Software Coding' met for 16 weeks, and two hours a week to study theory and practice for computational thinking by experiencing Python programming language. Classes were not focused on programming but focused on problem solving with computational thinking.

3.2 Effectiveness of Education

Fig. 1 shows the scores of computational thinking before and after the semester. The questionnaire after the semester was divided into two and analyzed. The scores were analyzed by classifying students with positive cognition about the class and those who with negative cognition. The gray bar graph shows the computational thinking scores of students with positive cognition while the orange bar graph shows for negative cognition.

In both cases, scores of computational thinking were improved. You can see that the scores of students with positive cognition have risen much higher than negative cognition. Table 1 show the ratio of improvement of computational thinking showing comparison between positive cognition and negative cognition.



(Figure 1) 5-point Likert Scale of CT

(Table 1) Ratio of Improvement

Factor	Computational Thinking	Negative (%)	Positive (%)
CT01	1. Reformulating	19.54212	43.13666
CT02	2. Recursive Thinking	33.96118	41.53282
CT03	3. Data as Code	23.91146	38.09752
CT04	4. Simplifying	28.20629	33.74853
CT05	5. Abstraction	18.59504	27.72915
CT06	6. Decomposition	24.18119	29.28822
CT07	7. Pre-fetching	19.23077	24.74765
CT08	8. Resource Sharing	2.112287	17.05389
CT09	9. Heuristic Reasoning	1.402966	21.45224
CT10	10. Algorithmic Thinking	16.25022	27.33279
CT11	11. Data Collection	6.445536	18.86674
CT12	12. Data Analysis	1.392597	15.48705
CT13	13. Data Representation	0.878123	19.86537
CT14	14. Simulation	8.46187	22.97098
CT15	15. Parallelism	5.179477	24.84804
CT16	16. Problem Cognition	7.500537	20.22421
CT17	17. Pattern Recognition	4.24939	20.28614
CT18	18. Logical Thinking	11.72749	23.8608
CT19	19. Calculating Ability	3.545071	19.68719
CT20	20. Spatial Perception	12.33381	18.05841

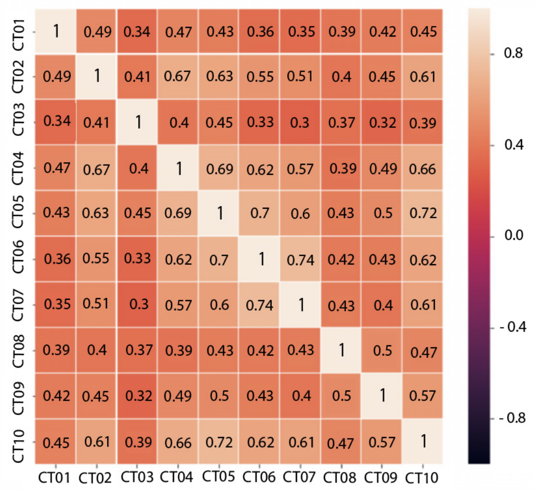
The scores for all factors are increased, and it shows the effectiveness in education. It can be seen that computational thinking can be improved through software education. Besides positive and negative cognition, some elements went up a lot but some went up a little.

The questionnaire was conducted for the students of the social sciences college in order to analyze the level of computational thinking. The lowest score among 20 factors was the reformulating at the beginning of the semester, while the lowest score was the parallelism after the semester. These scores can be used to strengthen the specific factors according to the major field of the students by analyzing the relation among the factors.

The lowest increased factor regardless of cognition was data analysis followed by resource sharing. It indicates which computational thinking should be more focused on at the class for improvement of the effectiveness of education to enhance the level of computational thinking.

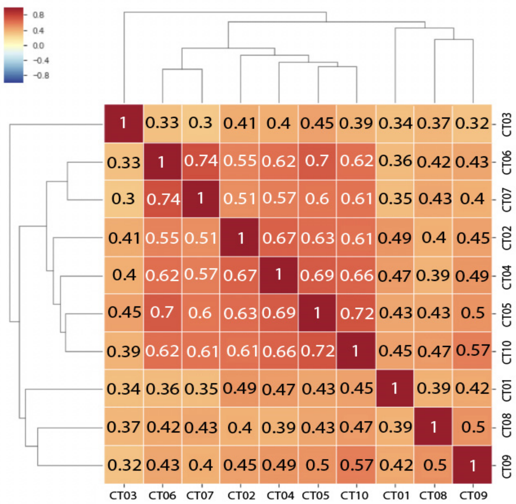
3.3 Analysis of Survey Data upon Wing's CT

Ten factors of computational thinking from the survey factors were extracted from the study of Professor Wing. To analyze these factors, correlation analysis was applied to evaluate the strength of relationship among these factors. Python packages of pandas, matplotlib, and seaborn were used in jupyter lab. Fig. 2 shows the correlation analysis among sub-factor for ten factors of computational thinking that Professor Wing suggested.



(Figure 2) Heat map for Wing's Suggestion

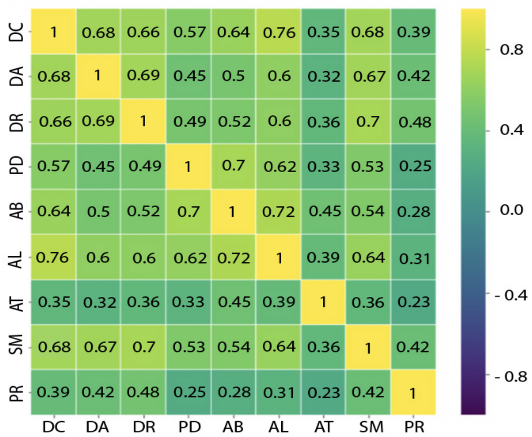
In order to check the detailed associations, a cluster map was drawn in Fig. 3. There is a close relationship between abstraction which is factor CT05 and algorithmic thinking which is factor CT10 those are the most important factors in computational thinking. Decomposition which is factor CT06 and pre-fetching which is factor CT07 are formed a clustering. It means after breaking down into smaller unit by decomposition, one can prepare what one needs to solve a problem by applying pre-fetching. Fig. 2 is verified that clustering is semantically well formed.



(Figure 3) Cluster map for Wing's Suggestion

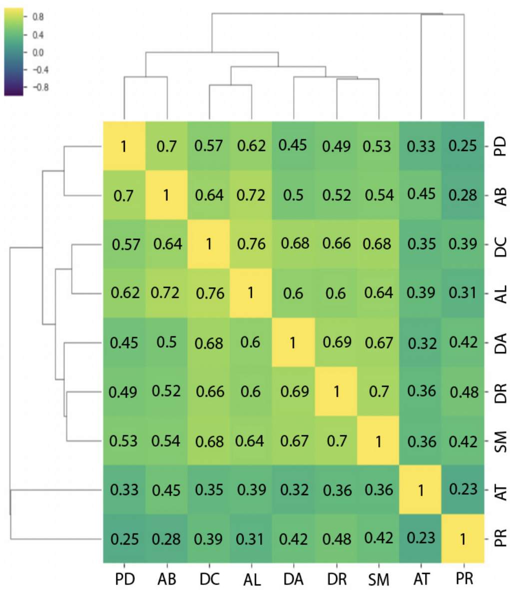
3.4 Analysis of Survey Data upon ISTE's CT

Computational thinking factors that ISTE proposed can be grouped into three categories which are data, design, and coding. For category of 'data', sub-factor decomposed into data collection, data analysis, and data representation. For category of 'design', sub-factor decomposed into problem decomposition, abstraction, and algorithmic thinking. For category of 'coding', sub-factor decomposed into automation, simulation, and parallelism. To analyze the relationship among the nine factors, heat map has been plotted in Fig. 4.



(Figure 4) Heat map for ISTE's Suggestion

To verify the relationship among sub-factor more accurately, the cluster map is shown in Fig. 5. As you can see automation and parallelism are not fit into the cluster map. It is obvious that these two factors are performed by computers, not humans. It means that it corresponds to the factors included in the programming process but not to the computational thinking of humans.



(Figure 5) Cluster map for ISTE's Suggestion

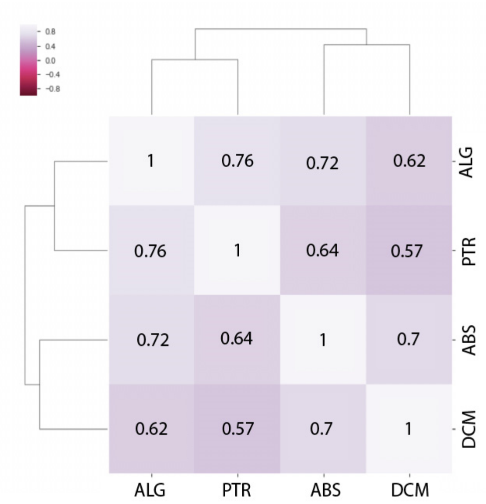
3.5 Analysis of Survey Data upon Google's CT

Google proposed four factors for computational thinking which are decomposition, pattern recognition, abstraction, and algorithm. To analyze the relationship among these four factors for computational thinking, heat map has been plotted in Fig. 6. As you can see the relationship among these four factors are very close.

The cluster map is shown in Fig. 7. It indicates that algorithm and pattern recognition are clustering together, and abstraction and decomposition are clustering together. We can interpret this phenomenon as follows. If we can catch the main concept of the problem, we can extract the rule from the problem and the rules can be defined.

(Figure 6) Heat map for Google's Suggestion

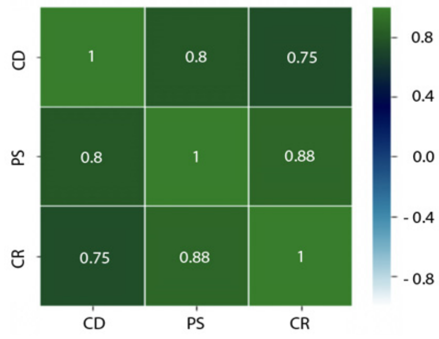
On the other hand, when we can apply abstraction toward the problem, we understand the key concept of the problem and we can divide the problem into smaller parts.



(Figure 7) Cluster map for Google's Suggestion

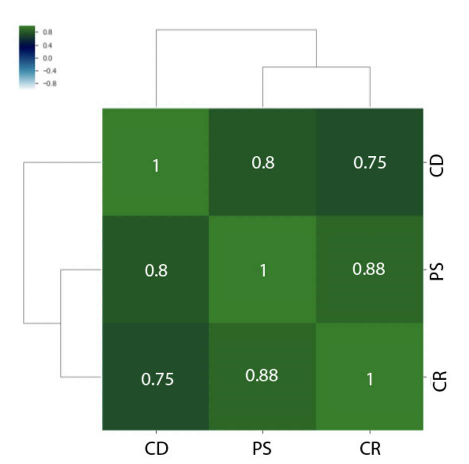
4. Result

To analyze the result of this study, we applied computational thinking as an independent variable and classified three dependent variables which are CD, PS, and CR. CD stands for coding which is software development skill, PS stands for problem solving, and CR is creative thinking. The heat map for these three dependent variables is shown in Fig. 8.



(Figure 8) Heat map for Result

The cluster map of three dependent variables are shown in Fig. 9. As a result, we found that enhancing computational thinking improves problem solving(PS) and creative thinking(CR), which can improve software skills which means coding ability(CD).



(Figure 9) Cluster map for Result

5. Conclusion

To acquire software development skills to prepare the 4th industrial revolution, the first step is to start to enforce and learn computational thinking. This is the direction of proper software education.

In this study, we verified that computational thinking is the first priority for software education. By learning computational thinking, not just coding techniques, students can develop problem-solving skills and creative thinking ability.

It is necessary to extend this research to analyze the characteristics of each major fields of students and suggest ways to strengthen the computational thinking for each case.

Reference

[1] K. Kim, "Domestic Research Trends and Cases of University Education and Operation in the Era of the Fourth Industrial Revolution", Journal of Digital Convergence, Vol. 17, No. 8, pp. 15-26, 2019. https://doi.org/10.14400/JDC.2019.17.8.015

[2] J. Kim, "Effect of Computational Thinking on Problem Solving Process in SW Education for non-CS Major Student", Journal of Korea Multimedia Society, Vol. 22, No. 4, pp. 472-479, 2019. https://doi.org/10.9717/kmms.2019.22.4.472

[3] O. Han and J. Kim, "Examining the relationship between educational effectiveness and computational thinking in smart learning environment", Journal of Internet Computing and Services, Vol. 19, No. 2, pp. 57-67, 2018. http://dx.doi.org/10.7472/jksii.2018.19.2.57

[4] O. Han and J. Kim, "Correlation Analysis of Factors of Computational Thinking", The 14th Asia Pacific International Conference on Information Science and Technology, 2019. http://http://www.apicist.org/2019/

[5] M. Chung and J. Kim, "The Internet Information and Technology Research Directions based on the Fourth Industrial Revolution", KSII Transactions on Internet and Information Systems, Vol. 10, No. 3, pp. 1311-1320, 2016. http://dx.doi.org/10.3837/tiis.2016.03.020

[6] S. Ham, "Attitudes and Performance of Workers Preparing for the Fourth Industrial Revolution", KSII Transactions on Internet and Information Systems, Vol. 12, No. 8, pp. 4038-4056, 2018. http://dx.doi.org/10.3837/tiis.2018.08.027

[7] O. Han and J. Kim, "The Study on Correlation of Cognition on Software Education with Improvement of Computational Thinking", Journal of Internet Computing and Services, Vol. 20, No. 3, pp. 93-100, 2019. http://dx.doi.org/10.7472/jksii.2019.20.3.93

[8] J. Wing, "Computational thinking and thinking about computing", Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 366, No. 1881, pp. 3717-3725, 2008. https://doi.org/10.1098/rsta.2008.0118

[9] ISTE, "Computational Thinking: teacher resources second edition", The International Society for Technology in Education, pp. 1-69, 2011. https://id.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2

[10] Google, "What is computational thinking?". https://sites.google.com/site/ityear7bcc/home/computational-thinking/lesson-1-computational-thinking

[11] J. Kwon and J. Kim, "A Study on the Design and Effect of Computational Thinking and Software Education", KSII Transactions on Internet and Information Systems, Vol. 12, No. 8, pp. 4057-4071, 2018. http://dx.doi.org/10.3837/tiis.2018.08.028

◐ 저 자 소 개 ◑

한 옥 영(Oakyoung Han)
1985년 The University of Kansas 대학교 Computer Science 학과 (B.S. 학사)
1989년 California 주립 대학교 대학원 Computer Science 학과 (M.S. 석사)
1999년 한국과학기술원(KAIST) 대학원 전산학과 (박사수료)
2012년 성균관대학교 대학원 컴퓨터교육학과 (박사)
1989년∼1991년 삼성전자 컴퓨터부문 응용개발실
1991년∼1999년 덕성여자대학교 전산과장, 연구교수
2000년∼2020년 성균관대학교 컴퓨터교육과 겸임교수
2015년∼2020년 성균관대학교 성균SW교육원
2020년∼현재 성균관대학교 학부대학 조교수
관심분야 : 소프트웨어교육, 컴퓨팅사고력, 자료구조, 알고리즘, 인공지능, SW+인문, 인터넷중독, etc.
E-mail : oakyoung@skku.edu

김 재 현(Jaehyoun Kim)
1988년 성균관대학교 수학과 졸업(학사)
1992년 웨스턴일리노이 주립대학교 대학원 전산학과 석사
2000년 일리노이공과대학교 대학원 전산학과 박사
2014년∼현재 한국컴퓨터교육학회 부회장 / 감사
2010년∼2019년 한국인터넷정보학회 부회장
2020년∼현재 한국인터넷정보학회 회장
2002년∼현재 성균관대학교 컴퓨터교육과 교수
2016년∼현재 성균관대학교 성균SW교육원 원장
2019년~ 현재 성균관대학교 사범대학장
관심분야 : 객체지향 소프트웨어공학, 컴퓨터교육, Computer Based LET etc.
E-mail : jaekim@skku.edu