International Journal of u- and e- Service, Science and Technology
Vol.10, No.8 (2017), pp.59-72
http://dx.doi.org/10.14257/ijunesst.2017.10.8.06

The Influence of Python Programming Education for Raising Computational Thinking

Youngseok Lee[1] and Jungwon Cho[2,*]

[1] *The Faculty of Liberal Arts, Kangnam University,*
40 Gangnam-ro Giheung-gu, Yongin-si, Gyeonggi-do, 16979 South Korea.
[2] *Department of Computer Education, Jeju National University,*
102 Jejudaehakno, Jeju-si, Jeju-do, 63243 South Korea.
[1] *yslee38@kangnam.ac.kr,* [2] *jwcho@jejunu.ac.kr*

Abstract

To cultivate talented individuals who will lead societies in the future, where the convergence phenomenon of various disciplines will be emphasized, it is necessary to provide a liberal arts curriculum that can cultivate competencies acquired as a fusion talent in the liberalization process of universities. If most computer-related liberal arts courses are designed to teach and practice how to use software, students should be taught the skills to create software they need or to ask computer program developers. To do this, we need to find a way to effectively implement programming education, and to find an education model that can stimulate the students' interests and improve computational thinking through interest in programming. According to a designed education model, students were asked to use a pre- and post-questionnaire consisting of 12 questions to understand the changes in computational thinking while conducting Python programming lessons for one semester. As a result, there were no differences between the groups, however the Python programming teacher was able to identify that the study had an impact on their computational thinking

Keywords*: Computational Thinking, Python education, Liberal arts, Programming Education*

1. Introduction

In the era of the 4th Industrial Revolution, universities are providing education specialized in training software talents that are highly sought after by employers. In particular, software education in universities has been fully reorganized based on the demand of the field, and Software basic education has been strengthened for non-traders to nurture multi-talented individuals who have knowledge of other major fields, including Software knowledge. The direction of software education is to cultivate creativity while improving the cognitive power of computers, by understanding problem solving through computer based algorithms and repeating the process of presenting an acceptable outcome for a problem [1].

At present, the importance of universities' liberal arts education is continuously emphasized, where the focus is on having students develop their basic abilities to create, apply, and adapt knowledge [2]. Until now, however, universities' computer literacy education has mainly been focused on information literacy education[3]. But in recent years not only information literacy but also computing debugging has been considered core competencies that should be taught in computer education. Related research is

* Corresponding author.

ISSN: 2005-4246 IJUNESST

actively being performed and it is included in the core competence of information education in Europe, including the United States (US) and the United Kingdom [4].

Computational thinking refers to the understanding of problem solving, system design, and human behavior based on the basic concepts and principles of computer science [5]. It is the basic knowledge that future talents must possess. In particular, for the development of computing algorithms, it is necessary for majors who are not related to IT to be able to find innovative items in their fields of study, to overcome their fear of technology, and to develop the ability to communicate with IT professionals [6]. This computational thinking can be developed and improved through computer programming language education.

The types of programming languages are very diverse, but it is desirable to approach them as a programming language that is interesting for beginners, easy to learn, and capable of writing various applications. The Python programming language is one of the appropriate programming languages. Python is used not only as a pure programming language but also as an intermediate language to connect modules written in other languages [7, 8].

The Python base code is easy to learn and particularly useful for beginners because it uses simple graphics processing [9]. It is useful for developing applications, and because it has a feature-full language, it can be used for convergent education. A total of 27 of the top 39 schools in the US, which have the most recognized computer science majors according to the US News & World Report, have adopted Python and 8 of the top 10 schools have Python training [10]. Looking at the current status of these universities, we can see that Python will be the programming language that is actively used in various fields in the future.

Therefore, the goal of the paper was to try to understand Python computer programming for cultivating computational thinking ability in liberal arts education, and to determine how this education affects students' improvement of computational thinking.

2. Python Computer Programming Education

2.1. Programming Education

In Korea, programming education starts from the 3rd stage (elementary school 5th to 6th grades) and starts with "Understanding the program". In the "understanding of information processing" (high school), the student gains an 'understanding of program production process' and 'developing application software'. The specific learning goals are shown in Table 1 [1, 11].

Table 1. Learning Objectives in Understanding Information Processing

Step	Contents
Step 3 (9–12 years old)	○ Programming understanding and basics ▪ Can recognize the concept of programming. ▪ Can recognize the programming language and basic usage. ▪ A simple program can be written and executed.
Step 4 (13–15 years old)	○ I / O programming ▪ Text-based input / output programs can be created and executed. ▪ Graphical I / O components and their characteristics can be recognized. ▪ It is possible to create and execute a simple program using graphical I / O components.
Step 5 (16~18	○ Understanding the program production process

years old)	▪ Can recognize the program making process. ▪ Can describe what is in the output of the program. ▪ Can suggest several ways to develop good software at low cost. ○ Application software production ▪ Can identify the characteristics of various programming tools and choose programming tools that are appropriate for writing the application software. ▪ It is possible to create and execute simple application software according to program production process using various programming tools including a data management system.

There are two main types of languages used to teach programming. One is a Based programming language, such as graphical elements, scratch, and app inventor, and a typing-based programming language, which allows commands to be entered into the grammar structure, such as Python and the logo [11, 12]. In particular, Python offers an easy way to learn computer programming through an environment called RUR-PLE [13, 14].

2.2. University Programming Education Examples

To study the effects of computer programming education, a study was conducted with 80 students who took 12 weeks of Python and App Inventor courses from the first semester of 2016. These students were university freshmen enrolled in 2016, who took courses in liberal arts that were not related to a computer major. The entire course was intended to cover 100% Python, but due to the difficulties found by some students, the course focused on 80% Python and 20% App Inventor. The contents of the lecture covered in the class are shown in Table 2, including 7 midterm examinations and 15 final examinations.

Table 2. Python Computer Programming Lesson Plan

Week	Subject	Contents
1	Introduction to Python	Installing and Running Python
2	Basic operation	Numbers, arithmetic expressions, and variables
3	Input and output statements	Variable, input / output function
4	Input and output statements	Input / output function, String type
5	Data Type #1	Boolean, comparison / logical operator
6	Conditional statement #1	Conditional statement, If ~ else
7	Conditional statement #2	Nested conditional statement, If ~ elif ~ else
8	Mid-term exam	
9	Data Type #2	List type, list index
10	Loop statement # 1	While Statement
11	Loop statement # 2	For Statement
12	Perform specific tasks	Function
13	Perform specific tasks	Class, Turtle class
14	Project	Project
15	Term exam	

Due to the nature of liberal arts courses, there are difficulties in the lectures due to students in the department of language and arts and the faculty of arts and sciences. In order to attract students' interest, there are four apps that utilize Paintbrush, TTS, Making efforts to improve computational thinking. As a result, the responses of the students analyzed through questionnaires are shown in Table 3 (N=40).

Table 3. Student Satisfaction Survey

Item	Department of Humanities and Social Studies	Faculty of Arts and Physical Education
Satisfaction of programming learning	82.1%	73.8%
Class difficulty	80.7%	73.8%
Programming Interest	82.1%	69.0%
Intent to learn programming major	81.4%	66.9%

As can be seen from the results of the satisfaction survey, there was a significant difference in satisfaction although the students were taught similar material, according to their preferences. Therefore, it is necessary to understand the level and interest of the students at the beginning of the class and to suggest the appropriate class model, as shown in Table 4, and to study the influence of computational thinking through programming education in the second semester [15].

Table 4. Modified Python Computer Programming Lesson Plan

Week	Subject	Contents
1	Understanding RUR-PLE and Python	Understanding RUR-PLE and Python basics
2	Learner Diagnosis # 1	Built-in functions and conditional statements
3	Learner Diagnosis # 2	Variable, input / output function
4	Learner Diagnosis # 3	Data Type, Loop statement
5	Conditional statement #1	Conditional statement, If ~ else
6	Conditional statement #2	Nested conditional statement, If ~ elif ~ else
7	Perform specific tasks	Function
8	Mid-term exam	
9	Data Type Project	Data Type Project
10	Loop Project # 1	While Statement Project
11	Loop Project # 2	For Statement Project
12	Function Project	Function Project
13	Turtle Project	Turtle Project
14	Project	Project
15	Term exam	

After the lecture has started, it was difficult to move or rearrange the students, but the average learners' level of comprehension should be divided into upper, middle, and lower classes. In addition, if a student is in a lower level group, those with a good understanding can suggest additional problems.

3. Computational Thinking Ability Test

3.1. Survey sheet of Computational Thinking Ability

An examination sheet for measuring the ability of computational thinking was prepared by extracting it from the Seoul Education Research & Information institute [16, 17]. Based on the revised lecture plan, we conducted a pre-test at the beginning of the lecture and a post-test at the end of the lecture. Computational thinking ability can be divided into nine factors, including abstraction, algorithm and its procedure [18]. However, considering the fact that it was a computer programming class as part of a liberal arts course, it was modified using abstraction, algorithms, procedures, and automation. The test items were as follows.

1. What are the numbers in box 7 if you follow the rules below?

[Rule]

(1) Multiply the number in the box (box no. 1) by "2" and put it in the box (box no. 2).

(2) Add "1" to the number in the box (box no. 2) and put it in the box (box no. 3).

(3) Repeat the rules (1) and (2) in front of you and try to put numbers up to box 7.

Box No.	1	2	3	4	…	7
Value	1	2	3	?	…	?

① 15 ② 16 ③ 30 ④ 31

2. If you follow the rules below, what are the numbers in boxes 9 and 10?

[Rule]

(1) Add "2" to the number in the box (box 1), and put it in the box (box 3).

(2) Subtract "1" from the number in the box (box no. 2), and put it in the box (box no. 4).

(3) Repeat the rules (1) and (2) above to put the numbers in boxes 9 and 10.

Box No.	1	2	3	4	…	9	10
Value	4	9	?	?	…	?	?

① Box No. 9 : 10, Box No. 10 : 6 ② Box No. 9 : 12, Box No. 10 : 5

③ Box No. 9 : 14, Box No. 10 : 4 ④ Box No. 9 : 16, Box No. 10 : 3

3. When Min-chul drinks three cups of milk, Min-Cheol's brother drinks a cup of milk. When the Sooji drinks two cups of milk, Sooji's sister drinks one cup of milk. When Mincheol and Sooji drink twelve cups of milk each, how many Mincheol's brother and Sooji's sister will drink a milk?

① Min-Cheol's brother : 10, Sooji's sister : 6

② Min-Cheol's brother : 4, Sooji's sister : 6

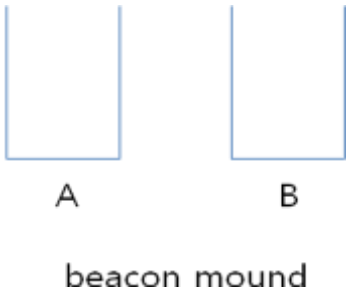
③ Min-Cheol’s brother : 6, Sooji's sister : 9

④ Min-Cheol’s brother : 8, Sooji's sister : 9

4. Taiho had raised six chickens. Six chickens gave birth to nine eggs a day. One day my father bought two chickens and raised eight chickens. How many eggs will be born each day for eight chickens?

① 6 ② 8 ③ 10 ④ 12

5. If a beacon mound is made up of two pieces, and each of these signal fire carry information according to the absence and presence of smoke, can the two beacon mounds carry a few pieces of information?



Ex) A : the presence of smoke, B : the absence of smoke

A : the presence of smoke, B : the presence of smoke

① 2 ② 4 ③ 6 ④ 8

6. What are the total number of letters in a string consisting of 6 letters that can be made using a total of 6 letters, four letters ‘A’ and two letters ‘B’?

Ex) AAABBA, AABABA, … etc.

① 2 ② 4 ③ 6 ④ 8

7. When the relation of heights of A, B, C and D is as follows, what is the right height of each person inequality?

A < B

C < D

D > A

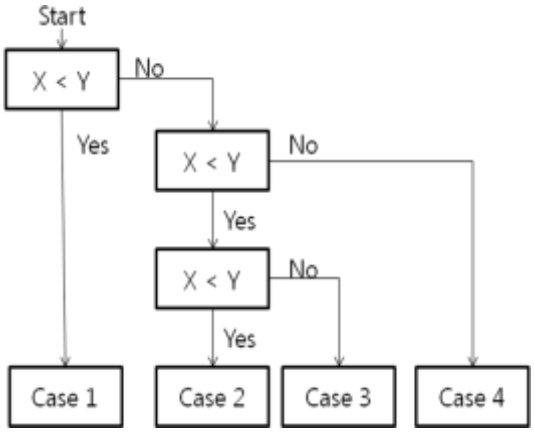
B < D

① B > A > D > C ② C > A > D > B

③ A > C > B > D ④ D > B > C > A

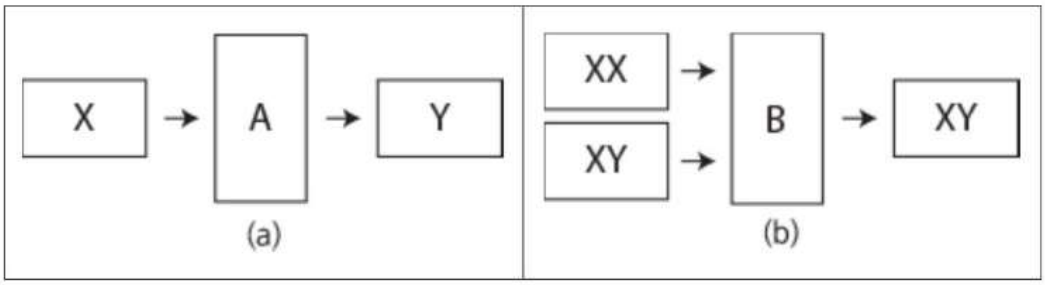
8. The following figure shows four possible paths according to the values of X, Y, and Z. Here, X, Y, and Z are not equal to each other. As a result of comparison of the

conditions, when the case 2 is reached, the relationship between X, Y and Z is correctly indicated by the inequality?

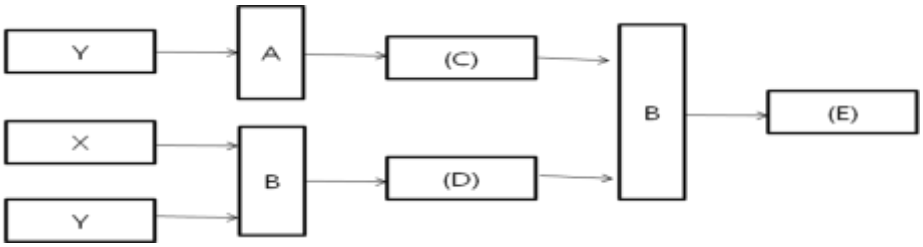


① X > Y > Z② Z > Y > X ③ Z > X > Y ④ X > Z > Y

[9–10] There are two boxes, A and B. Box A changes the letter × to y and y to x. Box B compares two characters and changes them to × if they are the same, or to y if they are different.

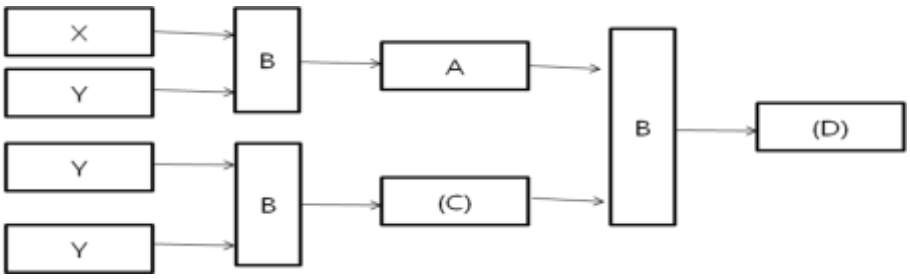


9. In the following circuit, what is the character to be put in the output (E)?



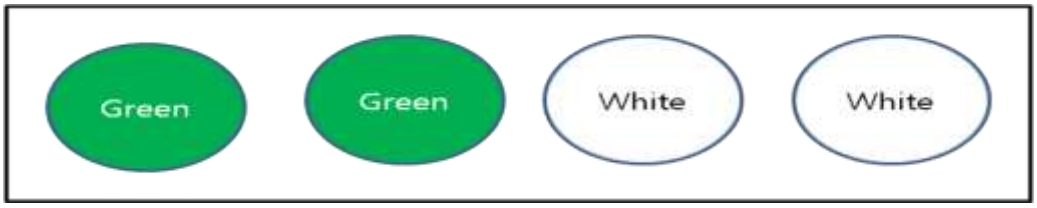
① X ② Y ③ XY ④ YY

10. In the following circuit, what characters would be in the output (D)?



① X ② Y ③ XY ④ YY

[11–12] Two green and white balls of the same shape and size are placed in the box, there are four.



11. What is the probability that a green ball will come out of a ball?

① 1/2 ② 1/4 ③ 1/6 ④ 1/8

12. What is the probability that a green ball will come out when one of the white balls is removed and another one is then taken out?

① 1/2 ② 1/3 ③ 2/3 ④ 1/8

3.2 Confidence Inspection

The questionnaires were used by the professors and the doctor who are computer education experts, and the items whose Cronbach's α value was more than or equal to 0.6 were used. The results of the reliability test are as follows.

Table 5. Total Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of items
.697	.714	12

Table 6. Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance	N o items
Item Means	.719	.367	.881	.514	2.403	.029	12
Item Variances	.176	.105	.249	.144	2.366	.003	12

The standardized item reliability analysis results of the pre-test also appeared to be more reliable than 0.6. The statistics for each item measuring computational thinking are as follows.

Table 7. Pre-test Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of items
.647	.674	12

Table 8. Item Statistics

	Mean	Std. Deviation	N
No1	.92	.267	105
No2	.88	.331	105
No3	.91	.281	105
No4	.73	.444	105
No5	.90	.308	105
No6	.41	.494	105
No7	.91	.281	105
No8	.84	.370	105
No9	.52	.502	105
No10	.50	.502	105
No11	.83	.379	105
No12	.56	.499	105

Table 9. Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance	N o items
Item Means	.694	.324	.876	.552	2.706	.031	12
Item Variances	.186	.110	.249	.139	2.269	.002	12

The results of reliability analysis of standardized items of post-test also showed more than 0.6. The statistics for each item are as follows.

Table 10. Post-test Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of items
.739	.749	12

Table 11. Item Statistics

	Mean	Std. Deviation	N
No1	.78	.416	105
No2	.80	.402	105
No3	.74	.439	105
No4	.88	.331	105
No5	.87	.342	105
No6	.32	.470	105
No7	.84	.370	105
No8	.56	.499	105
No9	.67	.474	105
No10	.41	.494	105
No11	.75	.434	105
No12	.70	.458	105

Table 12. Summary Item Statistics

	Mean	Minimum	Maximum	Range	Maximum / Minimum	Variance	N o items
Item Means	.694	.324	.876	.552	2.706	.031	12
Item Variances	.186	.110	.249	.139	2.269	.002	12

3.3 Analysis of Test Results

As for the comparative trends between computational thinking, with Python programming education, a t-test was conducted to analyze the congruity between the computational thinking. The findings are shown in Tables 13 and 14.

Table 13. Paired Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair1	PreSum	8.26	105	2.519	.246
	PostSum	8.99	105	2.268	.221

Table 14. Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Mean				
					Lower	Upper			
Pair1	Pre - Post	-.733	2.399	.234	-1.198	-.269	-3.132	104	.002

When comparing ability of computational thinking, the difference of mean value was 8.26 in pre-test and 8.99 in post-test; and the difference of standard deviation was 2.519 in pre-test and 2.268 in port-test. Resulting t-value was −3.132 in Paired test, indicating there was difference in mean values. So, the post-scoring score is higher than the pre-test score, and computational thinking is improved through Python programming lessons.

Considering the change of computational thinking, statistically significant conclusions can be drawn if treatment is used as covariate variable such as gender or grade. To do this, we conducted ANCOVA (Analysis of Covariance) analysis.

Table 15. Tests of Between-Subjects Effects

Dependent Variable : Sum

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	79.562[a]	3	26.521	4.777	.003
Intercept	630.833	1	630.833	113.622	.000
Grade	57.862	1	57.862	10.422	.001
Gender	8.279	1	8.279	1.491	.223
Period	18.900	1	18.900	3.404	.066
Error	1143.719	206	5.552		
Total	16841.000	210			
Corrected Total	1223.281	209			

a. R Squared = .065 (Adjusted R Squared = .051)

Since the p value of the mean score of the population is 0.003, the mean score is a statistically significant measure of computational thinking. Here, the p value (F value) of the Corrected Model is equal to the p value (F value) of the average number of points, as there is only one independent variable. Considering the above results, the question was asked, “Does computer programming teach you to improve your computational thinking skills by Grade or Gender?” If so, we can identify the covariate that has a connection between computer programming class and computational thinking. The p value = 0.003 for the whole model and the p value = 0.66 for the mean score are statistically insignificant. Therefore, it can be said that the current questionnaire method can improve the computational thinking ability as a result of the computer programming education, but it can be seen that it does not determine the difference based on gender.

4. Conclusion

In college, IT education is required to include computer based language programming, which can be interesting and learned regardless of major. However, students do not tend to have curiosity and are not interested in learning. To solve this problem, it is necessary to provide an easier method of programming and learning environment, and to perform various methods of systematic management. If we expand programming education opportunities based on the topics and contents of the lecture presented in this paper, it can improve students’ computational thinking abilities.

References

[1] Ministry of Science, ICT and Future Planning, Training software talent, http://www.msip.go.kr/webzine/posts.do?postIdx=175, **(2016)**
[2] S. H. Jin and S. B. S, Case Study and Needs Analysis on Convergence Education in Engineering Colleges, Journal of Engineering Education Research, Vol. 16, No. 6, **(2013)**, pp. 29-37.
[3] H. W. Lee, H. R. Min, K. W. Yi, A Study on the Improvable Proposal of General Education Curriculum of Engineering College - A Case of Seoul National University, Journal of Engineering Education Research, Vol. 11, No. 3, **(2008)**, pp. 24-32.
[4] H. Y. Jung, An Empirical Study on Information Liberal Education in University based on IT Fluency and Computational Thinking Concept, Journal of the Korea society of computer and information, Vol. 19, No. 2, **(2014)**, pp. 263-274.
[5] Wing, Jeannette M. "Computational thinking." Communications of the ACM , vol. 49, no. 3, **(2006)**, pp. 33-35.
[6] Begosso, L. C., Begosso, L. R., Gonçalves, E. M., & Gonçalves, J. R., "An approach for teaching algorithms and computer programming using Greenfoot and Python", In Frontiers in Education Conference (FIE), **(2012)**. pp. 1-6.
[7] Kyungmi Kim, Hyunsook Kim, A Case Study on Necessity of Computer Programming for Interdisciplinary Education, Journal of Digital Convergence, vol. 12, no. 11, **(2014),** pp. 339-348.
[8] Brennan, K., & Resnick, M., "New frameworks for studying and assessing the development of computational thinking", In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada, **(2012)**, pp. 1-25.
[9] Python, https://www.python.org/about/, **(2016)**.
[10] Philip Guo, Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities, BLOG@CACM, http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities/fulltext, **(2014)**.
[11] SungSoo Kim, "Design model for Python programming class utilizing scratch : focused on fifth grade elementary school", Thesis for Degree of Mater, Yousei University, **(2014)**.
[12] Tae-Ik Hyun, Jae-Hyuk Choi, Jong-Hee Lee. "Development of Teaching Model for Problem-solving methods and procedures; section in the 2012's revised Informatics curriculum." Journal of the Korea Society of Computer and Information , vol, 17, no. 8, **(2012)**, pp. 189-201.
[13] Andre Roberge, RUR-PLE Learing Python : Child's play with RUR-PLE!, http://rur-ple.xwmooc.org/RUR-PLE_Korean_ver_1_1.pdf, **(2014)**.
[14] Samsung junior software academy, https://my.juniorsw.com/class/03.html, **(2016)**.
[15] Edwards, S. H., Tilden, D. S., & Allevato, A. "Pythy: improving the introductory python programming experience", In Proceedings of the 45th ACM technical symposium on Computer science education, **(2014)**, pp. 641-646.
[16] Seoul education research & information institute, http://www.serii.re.kr, **(2016)**.
[17] Werner, L., Denner, J., Campe, S., & Kawamoto, D. C., "The fairy performance assessment: measuring computational thinking in middle school", In Proceedings of the 43rd ACM technical symposium on Computer Science Education, **(2012)**, pp. 215-220.
[18] Guo, P. J., "Online python tutor: embeddable web-based program visualization for cs education", In Proceeding of the 44th ACM technical symposium on Computer science education, **(2013)**, pp. 579-584.

Authors

Youngseok Lee received the B.S. and M. S. degree in Computer education from Seoul National University of Education, Seoul, S.Korea at 1998 and 2001, and graduated as Ph.D. degrees in Electronic Communication Engineering from Hanyang University, Seoul, S.Korea, and is currently enrolled the faculty of liberal arts in Kangnam University.

His research interests include smart learning, intelligent tutoring systems and web-based information systems.

Jungwon Cho received the B.S. degree in Information & Telecommunication Engineering from University of Incheon, Incheon, S.Korea at 1996, and earned the M.S. and Ph.D. degrees in Electronic Communication Engineering from Hanyang University, Seoul, S.Korea at 1998 and 2004, respectively. In 2004, he joined Jeju National University, Jeju, S.Korea, where he currently is a Professor at the Department of Computer Education. He was Vice-dean at the College of Education at 2011 and 2012. He also visited the Purdue University as a Visiting Scholar in 2007-2008 and the Salisbury University as a Visiting Professor in 2015`2015. He is an author of over 40 papers in refereed international journals and conference proceedings. His research interests include computer science (computing and software)education, information ethics, smart learning, intelligent system and multimedia. He is a member of the IEEE and the IEICE.

