Journal of Theoretical and Applied Information Technology
15th September 2017. Vol.95. No.17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THE EFFECTS OF THE COMPUTATIONAL THINKING-BASED PROGRAMMING CLASS ON THE COMPUTER LEARNING ATTITUDE OF NON-MAJOR STUDENTS IN THE TEACHER TRAINING COLLEGE

[1]YONGJU JEON, [2]TAEYOUNG KIM

[1, 2]Korea National University of Education, Department of Computer Education, 28173, South Korea

E-mail: [1]yyongju@naver.com, [2]tykim@knue.ac.kr

ABSTRACT

In these days, the software-integrated education is considered very important in both K-12 and undergraduate education. However, negative attitudes about software learning, such as lack of confidence caused by insufficient educational opportunities about computational thinking-based programming experience, could be a barrier to try software-integrated education for the non-majors in college. Unfortunately, only ICT skill-based education is provided for undergraduate students of the teacher training college in Korea, and we cannot find courses on computational thinking-based programming education in the university curriculum of liberal arts. Thus in this study, we developed a computational thinking-based programming course applicable to the liberal arts of the non-major education. Then we performed our experiment to both a controlled group with the traditional ICT skill-based course and an experimental group with our computational thinking-based programming course, and the paired samples t-tests were carried out. As a statistical result, the mean score of the experimental group is significantly higher on the most area of computer learning attitude. Thus, we concluded that the computational thinking-based programming class is more significant to foster affective elements of computer learning attitude such as superiority, self-confidence, interest, sense of purpose, accomplishment motivation and knowledge application than those of the traditional ICT skill-based class.

Keywords: *Computational Thinking Education, Computer Learning Attitude, Software-Integrated Education*

1. INTRODUCTION

The present age is called an era of computing, i.e., software-oriented society. In recent years, much attention has been paid to software-integrated technologies since real deliverables have been achieved via integration over various areas in academia, art and industries. Therefore, many nations and enterprises have emphasized computing technology as integration with academia, art and industrial area expecting visible achievements through investment and research [1-3].

This trend in recent years which is called the 4th industrial revolution has brought much attention to computing education around the world [4-12]. As a result, public and private educational institutions as well as private enterprises and organizations in South Korea have also provided various educational programs in the name of software education [11-12]. In the U.S. and the U.K., computing subject has already been the part of public education curriculum since 2013 and many other nations have also taken practical measures to participate in this trend for their national competitiveness. The core of this change in information and communication technology (ICT) education is to make students to learn how to produce computer programs or contents by themselves rather than learning how to use existing software [10]. This important educational change in computing will result in significant development of the national curriculum in the long term as well.

The guidelines of software education provided by the Ministry of Education of South Korea suggested mainly two types of software education [11]. The first type is software education as independent subject. Basically, this type of software education will be applied as informatics subject in middle and high schools by specialized teachers in computer education and will be applied as the part of practical course in elementary schools. The second type is software education as

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

integration with other subjects. This type of software education will require non-specialized teachers in computer-related subjects to have basic knowledge about computing in the long term.

The education curriculum used in the teacher training colleges in South Korea consists of teaching related subjects (education theory, teaching competency, education practice), major related subjects (curriculum of pedagogy, subject content education), and general educations (mandatory or optional) although some differences are found from university to university according to university-specific characteristics [13-15] (See Table 1). Among them, knowledge required in software education or contents related to programming competency can be provided with the subjects of general courses (mandatory) for their teaching competency.

Table 1: Curriculum of the Teacher Training Colleges in South Korea

Area		Courses
General pedagogy	Teaching theory	Educational pedagogy
	Teaching grounding	Practical skills, Student guidance
	Teaching practice	Field practice, Educational service
Major	Mandatory	Subject content education, Subject pedagogy education
	Selective	
Liberal arts	Mandatory	General courses
	Selective	Free courses

The general education subjects related to software education in the teacher training college have only focused on capabilities how to utilize ICT skills up until now. However, the future direction of software education shall extend general education subjects related to software education where computational thinking (CT) power is melted, and further cultivate affective motivation such as positive perspective and attitude by which prospective teachers can attempt software-integration education through successful experiences.

Thus, in this study we analyzed differences in learning attitude of prospective teachers who participated in lectures of two classes (each of them was a three-credit class for 15 weeks): one is an existing ICT skill-based class and the other is a new CT-based programming class among general education subjects in our university.

2. BACKGROUNDS

2.1 Future Human Talents and Software Literacy

P21 (Partnership for 21st Century Learning) in the USA predicted knowledge and technologies that will be required in daily living in the future society as shown in Figure 1.

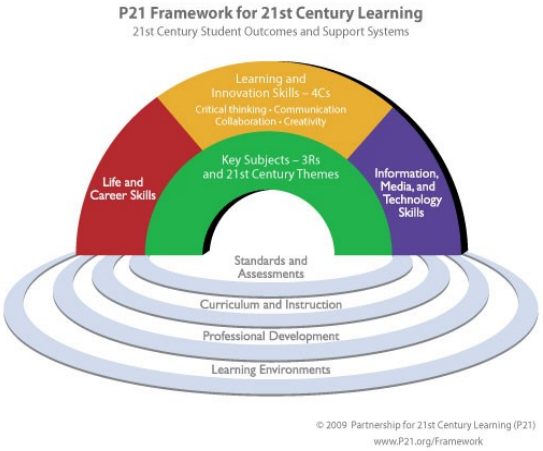

Figure 1: P21 Framework for 21st Century Learning [16]

The knowledge and technologies required in the future society are divided into four areas: life and career skills, key subjects and the 21st century themes, learning and innovation skills, and information, media and technology skills. Among them, *information, media and technology skills* refer to capabilities to deal with various types of data by utilizing software and create value-added or new knowledge through computer programming [16].

The DeSeCo (Definition and Selection of Key Competencies) project by the Organization for Economic Cooperation and Development (OECD) (2005) suggested three competencies for core competitiveness in the 21st century: competency of using tools interactively, competency of social interaction in different groups, and competency of managing own life autonomously (See Figure 2).

Among the three competencies, *competency of using tools interactively* has the following sub-capabilities: utilization capability of various communication tools (language, symbols, and texts), utilization capability of knowledge and information interactively, and utilization capability of new technologies [17], hinting that knowledge and competency related to software are essential for future talented persons to have competitiveness.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

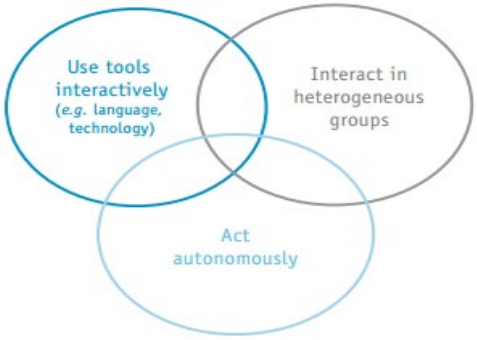

Figure 2: OECD DeSeCo Key Competencies [17]

Further, the World Economic Forum (WEF, 2017) describes the 16 skills required for students in the 21st century as shown in Figure 3.

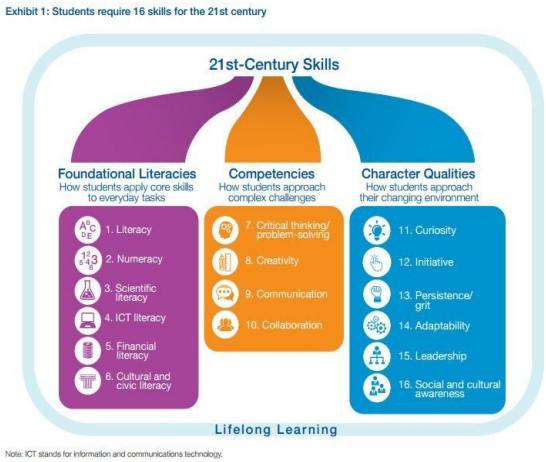

Figure 3: WEF 16 Skills for the 21st Century [18]

The future competencies of the WEF include six foundational literacies related to how students apply core skills in everyday tasks in the future society, four competencies related to how students approach complex challenges, and six character qualities related to how students meet the changing environment. Among them, many elements such as *ICT literacy, critical thinking, problem solving, creativity, communication, collaboration, curiosity, and adaptability* can be the skills associated with software education [18].

2.2 The Importance of a Positive Attitude in the Computing Education of Non-computer Majors

It is true that software competency and computational thinking ability are needed for future talented persons but learners who are not specialized in computer majors may feel unfamiliar or have difficulties in understanding educational contents on computing-related thinking skills.

Kelleher et al. (2003) argued that it is a great challenge for the beginners of programming education to learn how to apply structured solutions to the problems, to understand how a program is executed, and to learn rigorous grammar and commands that are named in unfamiliar forms [19]. Therefore, in order to meet such a challenge, affective aspect of programming education would be needed. Ben-ari (2001) asserted that grammar-based learning in programming education is less interesting for beginners and less likely to lead to the acquisition of actual developmental skills, and therefore it is important that learners should acquire an effective mental model of programming methods [20]. This mental model might be a functional aspect of the methodology, but it is also closely related to the affective aspect of programming education.

Kim (2015) analyzed difficulties facing at early education where computing thinking skills were applied by beginning learners among university students who did not major in computer-related subjects, and the result showed that novice learners had the most difficulty in understanding the concept of variables and lists, followed by designing and implementing ideas and selecting appropriate commands [21]. This argument also implied that the emotional aspects such as the attitude to learn by oneself and communication and collaboration among learners on contents and process are important for the programming education.

2.3 Instructional Model for the CT-based Programming Class

Considering above capabilities and competencies required in the current software-integrated society, the instructional model for the computational thinking-based programming class in our study is the Computational Thinking-based Creative Problem Solving (CT-CPS) model. Jeon and Kim (2015) developed the CT-CPS model which incorporates the computational thinking ability into the creative problem solving technique [22-24]. Our model is consisted of four stages; *problem identification and analysis*, *idea thinking*, *designing*, *implementation and evaluation* (See Table 2).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: The CT-CPS Instructional Model for Computing Education [20]

Stages	Teaching and Learning Activities
Problem identification and analysis	Motivation and introduction of tools used in the class
	Problem identification - Data collection on the subject to be problematized (interview, survey, field investigation, etc.) - Analysis and classification of collected data to be visualized (*tables, graphs, figures, etc.)
	Problem formulation - Reason to solve this problem (purpose), knowledge relevant to the problem, the value of the problem - Expression of a problem that should be ultimately solved (*writing, presentation, etc.)
Idea thinking	Coming up with diverse ideas to solve the problem using computing - Brainstorming, mind map, drawing, etc.
	Simplification of solution idea - Solvable level, abstraction
Designing	Design of the solution idea - Sketching the initial solution by each scene (*paper, presentation, etc.) - Making a logical flow chart
	Making a whole storyboard covering scenes and logics
Implementation and evaluation	Implementation of the solution idea - Making a implementation of the program based on a storyboard (* programming tools) - Debugging (along with teacher and peer)
	Presentation and demonstration (*presentation tools, etc.)
	Evaluation and feedback exchange - Teacher's observation and peer valuation

The CT-CPS instructional model allows students to solve problems through computational thinking while finding various problems through the activities of data collection, analysis, and expression in the problem identification stage. In addition, students can conduct group discussions and collaborative activities at the stages of idea thinking, design and implementation and evaluation. Therefore, these activities can help to develop a learner's self-efficacy and ability to collaborate.

3. METHODS

3.1 Course Design

Negative attitude and loss of self-confidence towards programming and lack of programming experiences are candidates of main hurdles to overcome software-fused education in fields and practice by prospective teachers majoring in various subjects. Thus, this study developed a program for general education course by utilizing web programming which can be relatively easily learnt and highly practical and aimed to achieve the following goals. First, prospective teachers can form positive attitude towards software education regardless of their majors. Furthermore, they can feel the sense of achievement and self-confident by providing an opportunity of planning of tasks and being successful. Second, it fosters competency of prospective teachers related to software education.

The class begins with the problem identification and analysis phase. As a starting process, the learning elements of CT such as data collection, data analysis, and data representation are practiced.

Secondly, in the idea thinking phase, students think about diverse ideas that can solve the problem at issue by making use of the relevant computing methods through a series of brainstorming, mind map, drawing and so on. The ideas are further simplified and abstracted so that it can be solvable at the learner's level.

Thirdly, in the designing phase, solution ideas that were extracted in a simplified form are structuralized. In this stage, the flow chart and storyboard designing are taken place through the process of visual design and logical design at the learner's level.

Finally, in the implementation and evaluation phase, the designed solution is actually implemented using a programming language through the process of draft writing (individual) and debugging (along with a teacher and peers). Afterward, the students present and demonstrate their completed output and manufacturing process, followed by feedback exchange from teacher observation and peer evaluation (See Table 3).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: The Contents of the CT-based Programming Course

Week	Contents	
1	Introduction of computational thinking	
2	Web and developing environment	
3	Basic structure of web documents, concept of server and client	
4	Problem -based learning	HTML5 - text, paragraph, image
5		HTML5 - links, video, audio
6		HTML5 - table and form elements
7		Mid test - making web documents
8		CSS3 - selectors
9		CSS3 - properties and values
10		CSS3 - responsive and mobile web
11	CT-CPS project (Making a respon -sive website)	Choosing theme and data collection
12		Idea thinking and designing a struc-ture of web site
13		Coding the website based on own design
14		Debugging and completing
15		Presenting outcomes and feedback

3.2 Learning Environments of CT-based Programming Course

In the CT-based programming course, Apache-based web server software for the Windows 7 was used. Further, web server spaces for each student in this course were provided through the file transfer protocol (FTP) service for storing web page files related to the homepage. The Coda2 IDE for Mac and the Sublime Text3 IDE for PC were utilized as programming tools.

3.3 Experiments

In order to quantitatively evaluate the purpose of this study, we designed and conducted an experiment for measuring the effect of the CT-based programming education class program on computer learning attitude of non-computer majoring prospective teachers.

The participants in our experimental study were 110 non-majors whose majors were not computer-related subjects and selected randomly regardless of majors in our university, a comprehensive teacher training college located in Chungbuk in South Korea. The subjects were divided into two groups taking one of two courses provided by our university: 59 participants of the experimental group in which the CT-based programming class program was applied, and 51 participants of the controlled group in which the existing ICT skill-based class program was applied (See Table 4). A period of the course was 15-week and 45-hour (three hours per week) classes. A pre-test on computer learning attitude was conducted prior to the course and a post-test was done immediately after the course completion.

Table 4: The Number of the Participants of the study

Semesters	Number of participants	
	Controlled group	Experimental group
1st	20	16
2nd	20	18
3rd	19	17
total	59	51
	110	

3.4 Testing Instrument

A testing instrument of computer learning attitude used in this study is a test questionnaire to measure self-concept, attitude toward the curriculum, and learning habit in the curriculum developed by the Korean Educational Development Institute, and modified by Lee (2010) in accordance with computer subject, was employed [25]. An assessment scale is a five level scale: “strongly agree”, “agree”, “neutral”, “disagree”, “strongly disagree” and the score was given from five to one point. Negative questions (No. 3, 8, 13, 18, 23, 28, 33, and 38) were given with negative point reversely. Table 5 shows the question number and the number of questions in each assessment scale.

Table 5: The Subsections of the Testing Instrument.

Subsections		Question No.	Questions
Self-concept for the subject	Superiority	1, 9, 17, 25, 33	10
	Self-confidence	4, 12, 20, 28, 36	
Attitude towards the subject	Interest	2, 10, 18, 26, 34	15
	Sense of purpose	5, 13, 21, 29, 37	
	Motive of accomplishment	7, 15, 23, 31, 39	
Study habits for the subject	Attention concentration	3, 11, 19, 27, 35	15
	Self-learning	6, 14, 22, 30, 38	
	Knowledge application	8, 16, 24, 32, 40	
Total			40

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The reliability of the test questionnaire which was done by Seo (2008) with high school students are as follows. The reliability coefficient of self-concept questions had a Cronbach alpha of .887, the reliability coefficient of questions about the attitude towards the subject had a Cronbach alpha of .745, and the reliability coefficient of questions about study habits for the subject had a Cronbach alpha of .729. In this study, the reliability coefficient measured with 110 students participated in our survey had a Cronbach alpha of .932, which was very high.

A difference in mean values between two groups in the post-test was analyzed via two-independent samples t-test and comparison of mean difference between pre and post-test within a group was analyzed via two-dependent samples t-test. The statistical software used in result analysis was IBM SPSS Statistics 21.

4. RESULTS

4.1 Homogeneity Test

In order to conduct a test of homogeneity between the controlled and experimental groups, results of the pre-test about computer learning attitude were analyzed using the independent-samples t-test. As a result, it showed p>.05 in all areas, which are presented in Table 6, indicating two groups were homogeneous in terms of computer learning attitude.

Table 6: The Results of the Pre-tests between Groups (p<.05, C.G.: Controlled group, E.G.: Experimental Group)

Subareas		Group	N	M	SD	t	p
Self-concept for the subject	Superiority	C.G.	59	3.09	.511	-.411	.682
		E.G.	51	3.12	.542		
	Self-confidence	C.G.	59	3.13	.486	.919	.360
		E.G.	51	3.05	.476		
Attitude towards the subject	Interest	C.G.	59	3.28	.354	.293	.770
		E.G.	51	3.26	.545		
	Sense of purpose	C.G.	59	3.60	.454	1.131	.260
		E.G.	51	3.48	.628		
	Motive of accomplishment	C.G.	59	3.29	.343	.773	.441
		E.G.	51	3.22	.560		
Study habits for the subject	Attention concentration	C.G.	59	3.34	.484	.974	.332
		E.G.	51	3.24	.541		
	Self-learning	C.G.	59	2.74	.496	-.066	.947
		E.G.	51	2.75	.555		
	Knowledge application	C.G.	59	3.25	.389	-.534	.594
		E.G.	51	3.29	.549		

4.2 The Results of the Post-tests between Groups

The post-test results about computer learning attitude of the controlled and experimental groups were analyzed via the independent-samples t-test (p<.05), which are presented in Table 7.

Table 7: The Results of the Post-tests between Groups (p<.05, C.G.: Controlled group, E.G.: Experimental Group)

Subareas		Group	N	M	SD	t	p
Self-concept for the subject	Superiority	C.G.	59	3.58	.517	-4.962	.000*
		E.G.	51	4.05	.425		
	Self-confidence	C.G.	59	3.35	.455	-4.028	.000*
		E.G.	51	3.71	.430		
Attitude towards the subject	Interest	C.G.	59	3.45	.384	-4.768	.000*
		E.G.	51	3.82	.403		
	Sense of purpose	C.G.	59	3.64	.363	-2.722	.008*
		E.G.	51	3.85	.416		
	Motive of accomplishment	C.G.	59	3.45	.398	-3.957	.000*
		E.G.	51	3.77	.436		
Study habits for the subject	Attention concentration	C.G.	59	3.34	.396	-1.353	.179
		E.G.	51	3.48	.646		
	Self-learning	C.G.	59	3.31	.599	1.238	.219
		E.G.	51	3.16	.737		
	Knowledge application	C.G.	59	3.34	.423	-3.411	.001*
		E.G.	51	3.64	.477		

The test results showed that *superiority* and *self-confidence* in the section of self-concept for the subject and *interest*, *sense of purpose* and *accomplishment motivation* in the section of attitude towards the subject and *knowledge application* in the section of study habits for the subject had a significant difference between two groups. It indicates that our CT-based programming class for non-computer majoring prospective teachers had more positive effect on above elements than the existing ICT skill-based class.

4.3 The Results between the Pre and the Post-test of Each Group

The pre-post test results of the computer learning attitude of each group were analyzed via the dependent-samples t-test (p<.05), which are presented in Table 8 and Table 9.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: The Results between the Pre and the Post-test of the Controlled Group (p<.0)

Subareas		test	N	M	SD	t	p
Self-concept for the subject	Superiority	pre	59	3.09	.511	-5.048	.000*
		post	59	3.58	.517		
	Self-confidence	pre	59	3.13	.486	-2.607	.012*
		post	59	3.35	.455		
Attitude towards the subject	Interest	pre	59	3.28	.354	-2.387	.020*
		post	59	3.45	.384		
	Sense of purpose	pre	59	3.60	.454	-.252	.802
		post	59	3.64	.363		
	Motive of accomplishment	pre	59	3.29	.343	-2.280	.026*
		post	59	3.45	.398		
Study habits for the subject	Attention concentration	pre	59	3.34	.484	.088	.930
		post	59	3.34	.396		
	Self-Learning	pre	59	2.74	.496	-5.365	.000*
		post	59	3.31	.599		
	Knowledge application	pre	59	3.25	.389	-1.377	.174
		post	59	3.34	.423		

Table 9: The Results between the Pre and the Post-test of the Experimental Group (p<.0)

Subareas		test	N	M	SD	t	p
Self-concept for the subject	Superiority	pre	51	3.12	.542	-9.560	.000*
		post	51	4.05	.425		
	Self-confidence	pre	51	3.05	.476	-6.606	.000*
		post	51	3.71	.430		
Attitude towards the subject	Interest	pre	51	3.26	.545	-5.125	.000*
		post	51	3.82	.403		
	Sense of purpose	pre	51	3.48	.628	-3.248	.002*
		post	51	3.85	.416		
	Motive of accomplishment	pre	51	3.22	.560	-5.735	.000*
		post	51	3.77	.436		
Study habits for the subject	Attention concentration	pre	51	3.24	.541	-2.052	.046*
		post	51	3.48	.646		
	Self-learning	pre	51	2.75	.555	-2.887	.006*
		post	51	3.16	.737		
	Knowledge application	pre	51	3.29	.549	-3.265	.002*
		post	51	3.64	.477		

The pre-post test results on computer learning attitude in the controlled and experimental groups showed that the means of the controlled group are improved in the following sections significantly such as *superiority*, *self-confidence*, *interest*, *motive of accomplishment* and *self-learning* while the experimental group had significant improvements in the means of *all* sections. It can be interpreted that the ICT skill-based class can also be positive to computer learning attitude of prospective teachers. However, the means of our CT-based programming class are improved in all sections higher than those of the traditional ICT skill-based class program.

4.4 Participants' Feedbacks and the Outcomes

The participants in the experimental group left feedbacks about the class anonymously after completing the 15-week course and some of them are as follows:

"This is the most helpful general education lecture that I have ever had. It is really good lecture that can be utilized in a class as well as in the outside of class."

"Developing a web site was very interesting to me because not only it required all details from contents, frame, and design, which was the same as creating a presentation file, but it also required coding, which was more complicated and challenging than presentation preparation. It was a pity that I did not make a responsive web site but I will develop a responsive web in the future when I am given an opportunity to make another web site. "

"Since the lecturer explained unfamiliar contents easily, it was really interesting to me to learn about HTML5 and CSS. After I become a teacher later, I would like to utilize my major to provide a useful software education."

"I was worried about HTML or CSS because I had no idea about what they were. However, it was a meaningful lecture to me because it made me attractive to what I didn't know before and learned more about them."

The deliverables submitted by the participants as individual project assignments after completing the course in a semester from the CT-based programming class applied in this study are shown in Figure 4 and Figure 5.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

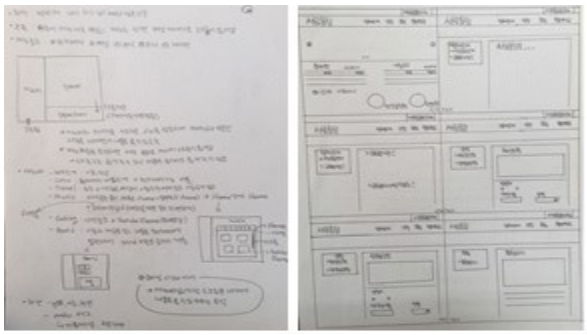

Figure 4: An Example of the Webpage Designs of the Participants

Figure 5: An Example of the Webpage Outcomes of the Participants

4.5 Interpretation of Research Outputs

The comparison results of the pre-test and the post-test within a group showed that a group that applied the ICT-based general education class program had statistically significant improvements of study subjects on some sections in computer learning attitude such as all sub-elements of self-concept for the subject, interesting and motivation in the attitude section toward the subject, and self-learning in the study habit section about the curriculum while a group that applied the CT-based programming general education class program had statistically significant improvements on all sections in computer learning attitude. Furthermore, the comparison results of post-test between two groups showed that a group that applied the CT-based programming general education class program revealed a statistically significant difference in all sections of computer learning attitude except for attention and self-learning more than a group that applied ICT -based class program.

The above results can be interpreted that both of the ICT-based class and the CT-based programming class promoted self-concept for the subject such as superiority and self-confidence in computer learning attitude of non-computer majoring prospective teachers. In particular, the CT-based programming class was more significant to foster affective elements such as a sense of purpose about computer education and learning skill application in the curriculum through experiences related to computing education and programming than the ICT-based class. In conclusion, it implies that the CT-based programming class was more effective than the ICT-based class.

Further, from the qualitative viewpoint, feedback from the participants shows that they have developed an attitude to apply computing to real life problem solving, and have raised interest and confidence in computing.

5. CONCLUSION

This study aimed to search for a method to foster affective competency, that is, positive viewpoints and attitudes about computing for non-computer major prospective teachers who will teach future generation. To achieve this, existing ICT-based course and computing thinking skill-based programming course (each course had three-credit for 15 weeks) selected from general education courses of a university which was a teacher's training college in Korea, were opened for three semesters and differences in computer learning attitude of students who participated in the courses were analyzed thereby investigating the changes in the prospective teachers.

Most teachers in the education field agree that providing computing education is justified and education of fusing with computing and existing subjects is needed now. However, most non-computer majoring teachers lack an opportunity or no opportunity to learn computing competency and specialty, which is why they are reluctant to attempt fusion of existing subjects with computing. Unfortunately, existing non-computing majoring teachers in the field did not have an opportunity to take computing education so they have to fill the lack through self-learning or related training. However, prospective teachers who will become a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

teacher in the future shall be equipped with basic computing education and related competency from the training stage. To achieve this, it is necessary to provide a general education course that can experience not only existing ICT-based general education but also introducing programming elements or computing thinking skills.

Our future research direction is to develop more CT-based training programs for non-computer majors. This study is expected to be utilized as foundational data to provide a general education course that can foster computing competency from the training stage for non-computer majoring prospective teachers.

This study has the following limitation. The participants of our experiment were 110 university students of a university in Korea, which might be limited in generalization of research results. However, the results of this study are quantitatively validated by comparing the results on the computer learning attitude before and after the class and qualitatively verified by the contents of the participants' testimonies and outputs to secure validity.

ACKNOWLEDGEMENTS

The authors thank the support of Korea National University of Education. The corresponding author is Taeyoung Kim. (tykim@knue.ac.kr)

REFERENCES:

[1] Nobel Prizes and Laureates, "The Nobel Prize in Chemistry 2013", 2013, http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013

[2] Google Deep Mind, "Solve Intelligence, Use it to Make the World a Better Place", 2016, https://deepmind.com

[3] New York Times, "Mercedes's Self-driving Car", 2015, http://nyti.ms/19Dm6Si

[4] Wing, J., "Computational Thinking", *Communications of the ACM,* Vol. 49, 2006, pp. 33-35.

[5] Wing, J., "Computational Thinking and Thinking about Computing", *Philosophical transactions of the Royal Society A*, Vol. 366, 2008, pp. 3717-3725.

[6] Barr, V. and Chris, S., "Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?" *ACM Inroads archive*, Vol. 2, 2011, pp. 48-54.

[7] CSTA, "CSTA K-12 Computer Science Standards Revised 2011", 2011, https://csta.acm.org/Curriculum/sub/K12Standards.html

[8] CSTA & ISTE, "Computational Thinking Leadership Toolkit 1st Edition", 2011, http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershiptToolkit-SP-vF.pdf

[9] Department for Education in U.K., "The National Curriculum in England", 2013, https://www.gov.uk/government/collections/national-curriculum

[10] Steve, M., "New National Curriculum to Teach Five Year Olds Computer Programming", 2014, http://www.techweekeurope.co.uk/news/national-curriculum-ict-education-computing-121214

[11] Ministry of Education in South Korea, "Guidelines for Software Education", 2015, http://www.moe.go.kr

[12] Ministry of Education in Korea, "The National Curriculum Revised in 2015", 2015, http://www.ncic.go.kr

[13] Seoul National University, "Curriculum of SNU 2014 (College of Education)", 2014, http://www.snu.ac.kr/upload/edu_file/undergraduate_course_list.pdf?ver=2014-08-14

[14] Korea National University of Education, "Curriculum of KNUE", 2015, http://www.knue.ac.kr/icons/app/cms/?html=/home/ life_co1.html&shell=/index.shell:192

[15] Seoul National University of Education, "Curriculum of SNUE 2015", 2015, http://www.snue.ac.kr/education/index.do?sub_no=1&sub_one_dept_no=0&sub_two_dept_no=5

[16] Partnership for 21st Century Skills, "Framework for 21st Century Learning", 2009, http://www.p21.org

[17] OECD DeSeCo, "The Definition and Selection of Key Competencies", 2005, http://www.oecd.org/pisa/35070367.pdf

[18] World Economic Forum, "What are the 21st-Century Skills Every Student Needs?" 2016, https://www.weforum.org/agenda/2016/03/21st-century-skills-future-jobs-students/

[19] Kelleher, C and Pausch, R. "Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers," *ACM Computing Surveys*, 2005, Vol. 37 No. 2, pp.83-137.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[20] Ben-ari, M. "Constructivism in Computer Science Education," *Journal of Computers in Mathematics and Science Teaching*, 2001, Vol. 20 No. 1, pp. 45-73.

[21] Kim, S. "Analysis of Non-Computer Majors` Difficulties in Computational Thinking Education," *Journal of Korean Association of Computer Education, 2015, Vol. 18 No. 3, pp. 49-57.*

[22] Jeon, Y. and Kim, T. "The Understanding of Software Education by the Analysis of International Trends", *Proceedings of the Korean Association of Computer Education Conference,* 2014, Vol. 18 No. 2, pp. 137-142.

[23] Jeon, Y. and Kim, T. "The Development of the CT-CPS Framework for Creative and Integrative Software Education," *Korean Journal of Teacher Education*, 2015, Vol. 31 No. 3, pp. 67-88.

[24] Jeon, Y. "The Development and Application of a CT-CPS Instructional Model for the Software Education of New Curriculum," Ph.D. dissertation, Korea National University of Education, 2017.

[25] Lee, J., "The Effects of Girl-friendly Teaching-Learning Program on Software Learning Attitude and Achievement", Master's thesis, Korea National University of Education, 2010.