JOIV : Int. J. Inform. Visualization, 6(2) - June 2022 317-326

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

A Design and Application of Software Liberal Arts Course based on CT-CPS Model for Developing Creative Problem-Solving Ability and Learning Motivation of Non-software Majors

Hee Jung Park [a], Yong Ju Jeon [a,*]

[a] *Department of Computer Education, Andong National University, 36723 Andong, Republic of Korea*
*Corresponding author: *yyongju@anu.ac.kr*

***Abstract*— As the importance of computing education for nurturing computational thinking skills is emphasized in preparation for the era of the 4th industrial revolution, computing education for non-majors is also expanding in liberal arts education at universities. In this study, a software liberal arts course based on the CT-CPS model was designed and applied to non-software majored university students, and the effect on creative problem-solving ability and learning motivation were analyzed. The CT-CPS (computational thinking-based creative problem solving) model is an instructional model devised by fusing each element of computational thinking ability to the creative problem-solving stages. Creative problem-solving ability test paper and learning motivation test paper were used as test tools. Moreover, quantitative analysis through independent sample t-test and paired sample t-test and qualitative analysis through subjective responses were conducted. As a result of the study, it was verified that the software class applied with the CT-CPS model had a statistically significant effect on the creative problem-solving ability and learning motivation of non-software majors. In particular, compared to the control group, the experimental group showed significant changes in the motivational elements among the sub-factors of creative problem-solving ability and the self-efficacy factor among the sub-factors of learning motivation. In addition, it was confirmed through qualitative analysis that the software class to which the CT-CPS model was applied helped develop the problem-solving ability and learning motivation based on computational thinking through the process of discovering and solving problems on their own in real life.**

***Keywords*—Software education; software liberal arts; CT-CPS instructional model; creative problem solving; learning motivation.**

Manuscript received 26 Oct. 2021; revised 11 Jan. 2022; accepted 15 Mar. 2022. Date of publication 30 Jun. 2022.

I. INTRODUCTION

Recently, many countries worldwide recognized the importance of computing education to develop computational thinking that can creatively and converge to solve complex and diverse problems in the era of the 4th industrial revolution and are applying it in the national curriculum. In Korea, to meet this global demand, software education has been made mandatory in Korea through elementary and secondary education courses, and the liberal arts education of universities is also expanding software education for non-majors to cultivate computational thinking skills [1], [2].

Wing [3] emphasized that computational thinking ability is the literacy everyone should have and argued that computer science and programming education would be necessary for the future regardless of a student's major. In order to flexibly respond to rapid social changes and to nurture talents who can lead the future society, universities are trying to reorganize their curriculum and innovate in education to strengthen their competencies related to computational thinking. In particular, related studies suggested that non-majors should focus on improving their creativity and problem-solving ability based on their understandings of software in order to cultivate computational thinking [4]–[8].

In order to increase the creativity of college students in educational innovation, it is necessary to develop a class model that integrates various factors related to creativity, such as motivation and problem-solving ability improvement [8], [9]. Motivation means learning motivation, and motivation is the driving force behind creative problem solving, and it can be said to be a very important factor in creative problem-solving as it has the power to persist the problem-solving process [10].

Learning motivation positively affects academic achievement, adaptation in school, and personal career exploration and planning. In the case of college students, it was said that, rather than individual characteristics, if the professor possessed knowledge and skills in the specialized field and felt emotionally connected with the students, it had a major influence on learning motivation [11], [12].

In addition, creativity is a very important core competency required of college students, but at the same time, it is a representative competency that is difficult to cultivate. The creativity curriculum of Korean universities is making efforts to nurture talents suitable for the era of the 4th industry, but it is insufficient compared to the educational requirements and importance [13]. If the creative competency is effectively developed in the liberal arts education of the university, the university's competitiveness will be equipped for nurturing future talents [8].

However, as a required liberal arts education in many domestic universities, software basic education is still operated as the same curriculum regardless of students' majors and competencies [14], [15] or most software classes are operated as programming classes that focus on functions rather than the experience of thinking for problem-solving [6], [16]. The biggest problem is that classes are being conducted without students being motivated by the software curriculum [7], [8]. In addition, research related to creativity and computational thinking cultivation of non-software majors is very insufficient compared to studies on software education for elementary and secondary schools, which have already established themselves as essential education.

In this study, we designed and applied a software liberal arts course based on the CT-CPS model for college students who are non-software majors to analyze the effect on creative problem-solving ability and learning motivation. Through this, it will be possible to contribute to suggesting the direction of software class design and teaching-learning methods in college liberal arts education.

II. MATERIALS AND METHOD

A. *Studies Related to Software Liberal Arts Courses for Non-majors*

In the world's leading universities, software education is being expanded to target non-computer majors for computational thinking cultivation, attracting attention as a core competency of the future. An understanding of computer science and programming is important as a basic knowledge for college students [4], [5], [17]. This is being used as a good reference model for opening software courses for non-majors at domestic universities [4], [18], [19].

In the United States, Carnegie Mellon University and Harvard University are also actively conducting computational thinking education for non-computer majors [17]. In particular, Harvard University's 'CS50(Introduction to Computer Science) ' class has been running since 2007 to teach non-majors computer science and programming skills; the most popular course, with many people changing to the developer path after taking it [6], [17], [18].

A-Level (Advanced Level), a course for university entrance in the UK, can be seen as a process to prepare students for basic studies so they can study at university. Even at the internationally recognized A-Level, the subject 'Computer Science' has been applied since 2015 to develop problem-solving skills using computational thinking based on computer science principles and technical understanding [19], [20].

The Information Processing Society of Japan has developed a curriculum standard for 'general informatics (processing) education', a liberal arts subject for all university students. Through the recently revised J17-GE, it can be said that the goal was to manage the quality of education by nurturing the basic skills necessary for living in the times for college students without distinction between the arts and sciences [21], [22].

As such, it was found that many universities in the world recognized the importance of problem-solving skills using computational thinking cultivation necessary for university students to live in the present age, and they are rushing to expand and apply software liberal arts education to non-majors. In line with the global trend, software education for non-majors is establishing itself as a liberal arts basic education in many domestic universities and is gradually expanding, led by software-centered universities.

Looking at the research on software subjects for non-majors in liberal arts education at domestic universities, Park [7] analyzed that software education as a liberal arts subject at universities uses block programming that has easy grammar and can be easily handled even by non-majors. And as the goal of software liberal arts education for non-majors, improvement of creativity and problem-solving ability was suggested rather than acquiring software skills.

Kim [23] analyzed that the direction was changed from the traditional programming grammar and technique to computational thinking skills through a study on the software liberal arts compulsory subjects of college students. He also argued that subjects for improving computational thinking are suitable as software liberal arts compulsory subjects for college students, and as a tool, python, rather than a block-based tool is suitable for basic software education at universities despite its difficulties.

Jung [24] argued that through her research, "there are significant achievements in the development of basic software education through various studies, but there is insufficient research to measure the educational effectiveness intended by the software basic education designer". She also presented the results by analyzing logical thinking ability, creative potential, and logical potential items to analyze the educational effectiveness of algorithmic thinking-based software in basic education for non-major college students.

Shin [25] insisted that computational thinking-based software classes for learning motivation and effective learning operation should be derived with creative problem-solving ability and an interesting curriculum. In addition, Kim et al. [26] developed a rubric that evaluates students' creative output from a computational thinking perspective in programming classes. Song [27] confirmed that university's reflection-based software education had a positive effect on thinking ability and problem-solving, and sense of achievement. Jung [28] stated that, for non-major students at universities, various teaching and learning methods are needed to help students understand because software education has a low connection with major subjects.

As described above, many domestic and international universities have conducted and operated various courses to enhance the effectiveness of software education. However, it was found that there are insufficient educational programs to increase the learning motivation and creative problem-solving ability of non-majors based on computational thinking.

B. Creative Problem-Solving Ability and Learning Motivation

The meaning of creativity is defined as the characteristic of thinking of new things [29]. Guilford [30] and Basadur [31] asserted that creative thinking leads to new results since problem-solving produces a new response to a new situation. It has been argued that although research on creativity and problem-solving ability have been conducted respectively, creativity and problem-solving ability can be viewed as basically the same mental phenomenon.

In this study, the creative problem-solving ability was defined as the ability to seek various solutions to complex, ambiguous, or dangerous problems that can be faced in daily life or major and to solve them in the best way. A representative model related to creativity is the Creative Problem Solving (CPS) model, which was developed by Parnes, Treffinger, etc. The study was started by Osborn [32] that proposes creative problem-solving in the industry in 'Applied Imagination'

Creative problem-solving skills are important [33] due to some reasons as follows:

- It can help solve problems that occur in real life and solve future problems wisely.
- It can improve our lives.
- It can lead to harmonious development between the creative and critical parts emphasize.

Learning motivation is defined as the force that initiates learning and determines the direction, continuity, and intensity [34]. Looking at the definition of learning motivation by scholars, learning motivation is defined as the tendency to work hard because learning activities are considered valuable or to achieve the intended goals by recognizing learning activities as meaningful and valuable. Therefore, whether students actively participate in learning-related activities and tasks is also a question of motivation for learning [35].

Before software education, accurate information delivery and motivating process must be preceded. This is because soft education without motivation is difficult to expect a vision for convergence education and improvement of creativity and problem-solving ability [7]. In other words, if motivation is not induced in a teaching-learning situation, the direction is lost, and motivation is very important for learner behavior [36].

Lee et al. [36] analyzed the relationship between creativity, problem-solving ability, and motivation of college students and asserted that problem-solving ability and motivation significantly explain the creative ability and disposition of college students. In addition, by revealing the correlation between the three variables, it was suggested that an integrated perspective is required to develop teaching methods for the development of creativity and learning ability of college students in the future.

As described above, although software education effectively enhances creativity and problem-solving ability, verification of the empirical effect is required as it is difficult to confirm immediately. Also, it was confirmed that learning motivation has an important effect on enhancing the overall creativity of college students. Therefore, it is necessary to study various teaching and learning methods to cultivate the creative problem-solving ability and learning motivation of non-major college students.

C. CT- CPS Model

The CT-CPS (Computational Thinking-based Creative Problem Solving) instructional model is a model devised by fusing each element of computational thinking ability to the CPS stages. The CT-CPS model expresses the process of creatively solving real-life problems through computational thinking abilities and proceeds as a process of 'problem identification and analysis, idea thinking, design-implementation and evaluation' [37], [38], [39]. The characteristic of this model is that in the stage of problem recognition and analysis, various activities for data collection, analysis, and expression are carried out. The problem is found and solved through computational thinking. The CT-CPS model-based software class was measured to be effective in cognitive and affective areas such as creative problem-solving ability, metacognition, computer learning attitude, and learning motivation [37], [38], [39].

Afterward, Koo et al. [40] developed and applied the CT-CPS framework-based class content for elementary school informatics gifted students, saying that it is effective in improving creative problem-solving skills. Kim et al. [41] revealed that software class using data literacy was designed and applied based on CT-CPS model to have a significant effect on improving computational thinking and creative problem-solving skills of middle school students. Kim and Kim [42] developed and explored a living lab process-based software education program by applying the CT-CPS model to promote middle school students' creative problem-solving and computational thinking skills. In addition, Lee and Kim [43] designed a CT-CPS-based creative convergence class of information science using physical computing for high school students and confirmed that it has a positive effect on improving learners' convergence competency.

The CT-CPS model is widely used in software class design for computational thinking promotion, creative problem-solving, and convergence competency for students, and its effectiveness has also been proven. Therefore, in this study, a CT-CPS model-based software class was designed for non-majors as a class composition principle that creatively and actively implements solutions to real-life problems. The CT-CPS instructional model is shown in Table 1.

TABLE I
CT-CPS INSTRUCTIONAL MODEL [37]

Stage	Teaching-Learning Activities and Lear ning Materials (*)
Stage 1. Problem identification and analysis	• Motivation and introduction of tools used in the class • Problem identification – data collection on the subject to be pro blematized (interview, survey, field in vestigation, etc.) – analysis and classification of collected data to be visualized (* tables, graphs, figures, etc.) • Problematization – reason to solve this problem (purpose),

	knowledge relevant to the problem, th e value of the problem – expression of a problem that should be ultimately solved (* writing, presentation, etc.)
Stage 2. 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
Stage 3. Design	• 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
Stage 4. Implementation and evaluation	• Implementation of the solution idea – making 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 – teachers' observation and peer evaluation

D. CT-CPS based Software Liberal Arts Course Design

1) The direction of course design: The purpose of this study is to design a software liberal arts course to improve the creative problem-solving ability and learning motivation of non-software majors, and the course was designed with the following directions. First, in the first week before the full-fledged programming class, understanding the 4th industrial revolution and the relationship between each learner's major is considered. And in the second week, the necessity of software education is reviewed, and learning motivation is established. Second, until the time before the midterm exam, basic instruction on how to use the programming language was conducted so that a team project using the CT-CPS model could be carried out. Third, in the 'problem recognition and analysis stage', which is a first stage of the CT-CPS class model, various activities for data collection, analysis, and expression are induced so that the students can find and recognize real-life problems themselves, recognize them, and problematize them over the 2 weeks. At this time, the project plan should be submitted according to the CT-CPS stage for fairness of evidence and evaluation when conducting classes based on the CT-CPS model. Fourth, each class begins with the latest IT news or software technology related to each major to strengthen and maintain learning motivation. In addition, it has the potential to motivate by dealing with practical tasks, and provides a learner-centered learning environment by design.

2) Results of course design: The software liberal arts course for non-majors developed in this study is an App Inventor programming course with the theme of 'understanding and utilizing IoT web programming'. The course consists of 3 hours and 15 weeks and is designed so that students can find and solve problems on their own while performing various activities through the CT-CPS model. Table 2 shows the contents of the CT-CPS based step-by-step teaching and learning activities designed according to the direction of the course design.

TABLE II
A DESIGN OF SOFTWARE LIBERAL ARTS COURSE BASED ON CT-CPS MODEL FOR DEVELOPING CREATIVE PROBLEM-SOLVING ABILITY AND LEARNING MOTIVATION OF NON-MAJORS

Week	Steps of Course	Topics of Class	Teaching & Learning Activities
1-2	Confirmation of relevance to major and motivation for learning	Introduction to the course and understanding of the 4th industrial revolution	• Imagination becomes reality (watch the future school video) • How to survive in the era of the 4th industrial revolution, changes in occupations, and relevance to majors
		Understanding algorithms and block coding	• Necessity of software education and algorithms (sandwich algorithm video viewing), introduction to block coding [practice] how to make a simple app- talking cat app, introduce yourself app
3-7	Understanding programming concepts and practicing with real life examples	Understanding variables	• Understanding a app inventor variables and conditional statements [practice] variable use - calculator making app, simple pocket money management app
		Understanding conditional and looping statements	• Understanding app inventor conditional and looping statements [practice] creating an app using conditional and looping statements – a number guessing app, bmi calculation app, health care app
		Understanding simple functions and lists	• Understanding simple functions and lists in app inventor [practice] creating an app that uses functions – a shape drawing app, a grade management app
		Understanding internal databases	• Understanding app inventor's internal database [practice] creating an app that utilizes an internal database

				• Nonsense quiz or major related quiz app
			Understanding maps and web features	• How to design and expand the restaurant map screen, introduction of sensors and other functions [practice] a restaurant app near our school using a map
8	Midterm exam			
9	Mini projects		Understanding extensions,	• Understanding extensions, importing public data, google developer, and app registration [practice] today's weather app, bus timetable app
10-14	CT–CPS stages	Stage 1. Problem identification and analysis	Identify problems	• Forming teams, explaining the CT-CPS course • Data collection and visualization on the subject you want to problematize, [writing a plan work sheet]
			Problematize	• Explore the reasons for solving this problem • Express the problem that each team wants to solve (using text, presentation, etc.)
		Stage 2. Idea thinking	Explore ideas for solutions	• Thinking about solution ideas in various ways (brainstorming, mind map, sketching, etc.) • Simplify solution ideas, solvable levels, find key elements. [writing a work sheet]
		Stage 3. Design	Design of solution ideas	• Sketch apps/programs by scene (use paper, drawing, presentation, etc.) • Write a complete storyboard that synthesizes vision and logic (natural language, flowchart, etc.) [writing a work sheet]
		Stage 4. Implementation and evaluation	Implementation of solution ideas	• Try drafting an app/program based on the storyboard • Debugging
15	Final exam – presentation (presentation, etc.) and demonstration (evaluate together, feedback on what was good/improved)			

E. *The Hypothesis of the Study*

This study aims to design and apply a CT-CPS model-based software class for cultivating creative problem-solving ability and learning motivation in software liberal arts subjects for non-majors and then verify its effectiveness through analysis of the results. The hypotheses of the study are as follows.

- Hypothesis 1: The creative problem-solving ability of non-major students will be different depending on the teaching and learning methods.
- Hypothesis 2: The learning motivation of non-major students will be different depending on the teaching and learning methods.

F. *Subject and Period of the Study*

This study was conducted for first-year students who were non-majors taking the ‘understanding and utilizing IoT web programming’ course at A university located in Gyeongsangbuk-do. This study consisted of a total of 30 students, 14 in the control group and 16 in the experimental group, and the application period was 15 weeks from September 1, 2021, to December 17, 2021.

G. *Design of Experiments*

Before the experiment, to confirm the homogeneity between the experimental group and the control group, a creative problem-solving ability pre-test, and a learning motivation pre-test were performed. After the pre-test, the control group took a software class in the form of problem-based learning to solve a given problem, and the experimental group applied the CT-CPS model-based software class. At this time, the control group was given the subject of an app for college students, and the experimental group was given the process of finding a topic on their own in the stage of problem recognition and analysis. The post-test was conducted immediately after the class was over, and the design of this application is shown in Table 3.

TABLE III
DESIGN OF APPLICATION

G1	O1	X1	O3
G2	O2	X2	O4

- G1: Control group
- G2: Experimental group
- O1, O2: Pre-test (creative problem-solving ability test, learning motivation test)
- X1: Software Course of problem-based learning model
- X2: Software Course based on CT-CPS model
- O3, O4: Post-test (creative problem-solving ability test, learning motivation test)

H. *Tools of the Study*

Mixed-method research was applied to analyze the results of this study. Quantitative research tools used to analyze this study were the creative problem-solving ability test and the learning motivation test. Moreover, the project report was analyzed to reveal the qualitative results of this study.

1) Test Tool of Creative Problem Solving Ability: In this study, the ‘Simple Creative Problem-Solving Ability Test Paper’ developed by the MI Research Team (2004) of the Psychological Lab at Seoul National University was used [44]. This tool was developed based on the ‘Simple Creative Problem-Solving ability test development study’ (2001) published by the Korea Educational Development Institute to measure creative problem-solving ability. The sub-factors of the tool consist of 'knowledge in a specific area, the function of thinking, skill, and mastery', 'divergent thinking', 'critical and logical thinking and 'motivational elements. There are 20 items, 5 items for each sub-factor, and each item is presented on a 5-point Likert scale. Table 4 shows the sub-factors and number of items in the creative problem-solving ability test tool.

TABLE IV
SUB-FACTORS AND ITEMS OF THE CREATIVE PROBLEM-SOLVING ABILITY TEST TOOL

Sub-Factor	Question	Number of Questions
Knowledge in a specific area, function of thinking, skill and mastery	1, 2, 3, 4, 5	5
Divergent thinking	6, 7, 8, 9, 10	5
Critical and logical thinking	11, 12, 13, 14, 15	5
Motivational elements	16, 17, 18, 18, 20	5
Total	20	

2) Test Tool of Learning Motivation: For the learning motivation test, a test tool was modified by Kang and Kwon [45] based on the Motivation Strategies for Learning Questionnaire (MSLQ). One teacher and two educational engineering experts re-verified the validity of this test paper [10]. The sub-factors of the learning motivation test paper consist of 'Internal goals', 'External goals', 'Control of learning beliefs', and 'Self-efficacy'. There are 16 items, 4 items for each sub-factor, and each item is presented on a 5-point Likert scale. Table 5 shows the sub-factors and number of items in the learning motivation test tool.

TABLE V
SUB-FACTORS AND ITEMS OF THE LEARNING MOTIVATION TEST TOOL

Sub-Factor	Question	Number of Questions
Internal goals	1, 10, 13, 14	4
External goals	5, 7, 9, 16	4
Control of learning beliefs	2, 6, 11, 15	4
Self-efficacy	3, 4, 8, 12	4
Total	16	

3) Methods of Qualitative Analysis: In this study, in addition to quantitative analysis, it was attempted to additionally secure the validity of the research results through qualitative analysis. For qualitative analysis, keyword analysis was performed on the contents of the project performance results and impressions of the project reports written by the students, and the responses to class evaluation and suggestions were further analyzed.

The CT-CPS model-based software class was conducted as a team project class after understanding the basic grammar, and it consisted of a step-by-step project plan for the progress, a project report, and a presentation and evaluation stage. At this time, the project report form was used by modifying the form for project planning and construction for non-majors [46].

The sub-items of the report consist of the project's planning background, objectives, content of the project, promotion system of the project, results, and impressions. Table 6 shows the sub-areas and details included in the project report form.

TABLE VI
SUB FACTORS AND DETAILS OF PROJECT REPORTS

Sub-Factor	The Details
Planning background of the project	The necessity of the project, importance of the project, expected effect of the project
Objectives and content of the project	The goal of development, the content of development, flow chart
Promotion system of the project	Role division of team members, cooperation system of team member
Results and impressions of the project	Performance results of the project, problems in project execution, future improvement points of the project, impressions of project execution

III. RESULT AND DISCUSSION

A. Results of Pre-test

In order to confirm the homogeneity between the experimental group and the control group, an independent sample t-test was performed on the difference in the mean of learners in each group based on each pre-test. IBM SPSS version 26 was used for data analysis and processing in this study. Table 7 shows the results of the pre-test for creative problem-solving ability.

TABLEVII
RESULTS OF THE PRE-TEST FOR CREATIVE PROBLEM-SOLVING ABILITY

Sub-Factor	Group	M	SD	T	P
Knowledge in a specific area, function of thinking, skill, and mastery	Control group	2.81	.523	.009	.993
	Experimental group	2.81	.558		
Divergent thinking	Control group	2.76	.662	-.934	.358
	Experimental group	2.96	.543		
Critical and logical thinking	Control group	3.74	.346	1.712	.098
	Experimental group	3.53	.349		
Motivational elements	Control group	3.26	.454	-.756	.456
	Experimental group	3.40	.566		

(*p < .05)

After the creative problem-solving ability pre-test, the control and experimental groups had similar average values. The significance level of each factor in the two groups was higher than .05, indicating no significant difference between the groups. Therefore, it was confirmed that the control and experimental groups were homogeneous groups for creative problem-solving ability. Table 8 shows the results of the pre-test for learning motivation.

TABLE VIII
RESULTS OF PRE-TEST FOR LEARNING MOTIVATION

Sub-Factor	Group	M	SD	T	P
Internal goals	Control group	3.48	.567	.208	.837
	Experimental group	3.44	.602		
External goals	Control group	3.70	.502	.676	.505
	Experimental group	3.56	.574		
Control of learning beliefs	Control group	3.71	.448	-.273	.787
	Experimental group	3.77	.566		
Self-efficacy	Control group	3.23	.373	-.014	.989
	experimental group	3.23	.504		

(*p < .05)

In the result of the pre-test for learning motivation, it was found that the control group had a slightly higher average than the experimental group, but the significance level of each factor in the two groups was higher than .05, indicating no significant difference between the groups. Therefore, it was confirmed that the control group and the experimental group were homogeneous groups for learning motivation.

B. Results of Post-test

1) Comparison of Post-test Results Between Groups: After the experiment, a post-test was performed to confirm the study results and whether there were any significant differences between the control and experimental groups. Moreover, Table 9 shows the post-test results of creative problem-solving skills compared to the t-test.

TABLE IX
RESULTS OF POST-TEST FOR CREATIVE PROBLEM-SOLVING ABILITY

Sub-Factor	Group	M	SD	T	P
Knowledge in a specific area, function of thinking, skill and mastery	Control group	3.30	.636	-.563	.578
	Experimental group	3.45	.798		
Divergent thinking	Control group	3.37	.756	-.927	.362
	Experimental group	3.64	.807		
Critical and logical thinking	Control group	4.00	.748	-.600	.554
	Experimental group	4.15	.622		
Motivational elements	Control group	3.54	.861	-2.216	.035*
	Experimental group	4.14	.601		

(*p < .05)

As a result of the post-test on creative problem-solving ability, the average value of the experimental group was overall higher than that of the control group. As a sub-factor, the p-value of the 'Motivational elements' area was .035, which was below the significance level of .05, indicating a statistically significant difference. Therefore, it was confirmed that the creative problem-solving ability of students who took the CT-CPS model-based software class had a significant effect in the 'Motivational elements' area among the sub-factors of creative problem-solving ability compared to the students who took the problem-based learning type software class.

Table 10 shows the results of the comparison between groups of the learning motivation post-test.

TABLE X
RESULTS OF POST-TEST FOR LEARNING MOTIVATION

Sub-Factor	Group	M	SD	T	P
Internal goals	Control group	3.75	0.826	-.897	.378
	Experimental group	4.00	0.701		
External goals	Control group	3.73	0.811	-.007	.994
	Experimental group	3.73	0.883		
Control of learning beliefs	Control group	4.02	0.584	-.871	.391
	Experimental group	4.20	0.579		
Self-efficacy	Control group	3.30	0.827	-2.113	.044*
	Experimental group	3.89	0.695		

(*p < .05)

The average value of the experimental group was overall higher than that of the control group, and as a sub-factor, the p-value of the 'Self-efficacy' area was .044, which was below the significance level of .05, indicating a statistically significant difference. Therefore, we could not reject the second research hypothesis as a whole, but it was confirmed that the learning motivation of the students who took the CT-CPS model-based software class had a significant effect in the 'Self-efficacy' area among the learning motivation sub-factors compared to the students who took the problem-based learning type, software class.

2) Comparison of Pre- and Post-test Results Within the Groups: In order to examine the changes in creative problem-solving ability and learning motivation within each experimental group, the pre-and post-scores were compared, and Table 11 shows the results of the control group's paired sample t-test.

TABLE XI
COMPARISON OF PRE- AND POST-TEST RESULTS OF THE CONTROL GROUP

	Sub-Factor	Paired	M	SD	T	P
Creative Problem-Solving Ability	Knowledge in a specific area, the function of thinking, skill, and mastery	pre	2.81	.523	-2.192	.047*
		post	3.30	.636		
	Divergent thinking	pre	2.76	.662	-2.381	.033*
		post	3.37	.756		
	Critical and logical thinking	pre	3.74	.346	-1.142	.274
		post	4.00	.748		
	Motivational elements	pre	3.26	.454	-1.192	.255
		post	3.54	.861		
Learning Motivation	Internal goals	pre	3.48	.567	-1.005	.333
		post	3.75	.826		
	External goals	pre	3.70	.502	-.155	.879
		post	3.73	.811		
	Control of learning beliefs	pre	3.71	.448	-1.588	.136
		post	4.02	.584		
	Self-efficacy	pre	3.23	.373	-.300	.769
		post	3.30	.827		

(* : p< .05)

As a result of comparing the pre-test and post-test for each test in the control group, among the sub-factors of the creative problem-solving ability test, p-values were .047 and .0332 in the 'Knowledge in a specific area, Function of thinking, Skill and mastery' area, and the 'Divergent thinking' area, respectively, therefore, the significance level was less than .05, and a statistically significant change was observed, and there was no statistically significant difference in the sub-domain of the learning motivation test, but an increase in the mean value was observed in all areas. Table 12 shows the results of the paired sample t-test of the experimental group.

TABLE XII
COMPARISON OF PRE- AND POST-TEST RESULTS OF THE CONTROL GROUP

	Sub-Factor	Paired	M	SD	T	P
creative problem-solving ability	Knowledge in a specific area, function of thinking, skill, and mastery	pre	2.81	.558	-2.272	.038*
		post	3.45	.798		
	Divergent thinking	pre	2.96	.543	-2.387	.031*
		post	3.64	.807		
	Critical and logical thinking	pre	3.53	.349	-3.552	.003**
		post	4.15	.622		
	Motivational elements	Pre	3.40	.566	-5.021	.000**
		post	4.14	.601		
learning motivation	Internal goals	pre	3.44	.602	-2.563	.022*
		post	4.00	.701		
	External goals	pre	3.56	.574	-.729	.477
		post	3.73	.883		
	Control of learning beliefs	pre	3.77	.566	-2.671	.017*
		post	4.20	.579		
	Self-efficacy	pre	3.23	.504	-3.416	.004**
		post	3.89	.695		

(* : p< .05, ** : p< .01)

As a result of comparing the pre-and post-tests for each test in the experimental group, in all the creative problem-solving ability test sub-factors, p-values were .038, .031, .003, and .000, respectively, so the significance level was less than .05, indicating a statistically significant change. Moreover, among the sub-factors of the learning motivation test, 'Internal goal', 'Control of learning beliefs', and 'Self-efficacy' excluding 'External goal' had p-values of .022, .017, and .004, respectively, so the significance level was less than .05, and a statistically significant change was observed. In addition, the mean value was observed in the entire area of the experimental group.

The factors of each test are the comparison results between groups of the post-test and the comparison results of the pre-post tests of the experimental group. Table 13 shows the factors that showed statistically significant improvement. As a result of comparing the tests between groups, it can be seen that the CT-CPS model-based software course had a positive effect on the improvement of the 'Motivation factor' area of the creative problem-solving ability test and the 'Self-efficacy' area of the learning motivation test compared to the problem-based learning model software course.

C. Interpretation of Quantitative Analysis Results

TABLE XIII
COMPREHENSIVE TEST RESULTS FOR EACH TEST FACTOR

Test	Sub-Factor	Statistical Improvement of Comparison Results Between Post-Test Group	Statistical Improvement of Pre and Post Comparison Results of the Experimental Group
Creative Problem-Solving Ability	Knowledge in a specific area, function of thinking, skill, and mastery		○
	Divergent thinking		○
	Critical and logical thinking		○**
	Motivational elements	○	○**
Learning motivation	Internal goals		○
	External goals		
	control of learning beliefs		○
	self-efficacy	○	○**

(**: p< .01)

In addition to pre and post-test comparison results within the group, the CT-CPS model-based software course, through the process of finding and solving problems independently, showed improvement in all areas except for 'External goal'. In particular, it can be said that the areas of 'Critical and logical thinking', 'Motivational elements', and 'Self-efficacy' showed very big improvement

D. Results of Qualitative Analysis

In this study, subjective responses to project reports were analyzed for qualitative analysis of the effects of CT-CPS model-based software classes on creative problem-solving ability and learning motivation of non-major students. In addition, meaningful answers were further analyzed in the suggestion area of class evaluation and the subjective feedback area of the evaluation stage. Fig. 1 shows the visualization using keywords for easy recognition [47].

Fig. 1 Visualization of KeyWords

As a result of analyzing the qualitative responses of students who took software classes applying the CT-CPS model. However, it was difficult to set and implement topics to be solved in real life. The majority of opinions were that it was good to feel a sense of accomplishment by using the learned functions to solve problems individually and by successfully implementing the app. The following are student responses related to creative problem-solving ability.

- Student 1: I felt that the ability to think is the most important because the app is completed through the process of constantly thinking about and executing the blocks to create this function.
- Student 2: When I first encountered the subject, I felt a sense of pressure, but the fact that I was able to create an app myself through lectures and practice was surprising, and it was interesting to see how it works as I envisioned and built it. While doing the project, I realized that even a simple function had to go through more processes than I thought, and I could feel the computer accept and execute only commands.

In addition, they expressed their efforts and will to succeed in areas that could not be solved even after the project, and they wanted to develop more advanced major-related or major-related apps. And after that, it was confirmed that there was a motivation to learn the advanced software course. The following are responses related to learning motivation.

- Student 3: It was new to use only the apps that had already been made, and it was difficult to make the process by dividing roles with each other, but it was fun. It was a project that made me want to develop a better app by further developing it along with the major I will learn in the future.
- Student 4: I felt accomplished and proud when the project was gradually completed in the direction we wanted.

In another opinion, app inventor programming is suitable for improving the basic understanding of coding, and the project class allowed team members to share various ideas and foster a sense of collaboration. Moreover, it was found that other teams' planning intentions and creative ideas were impressive. Therefore, the software class based on the CT-CPS model had a positive effect on cultivating creative problem-solving ability and learning motivation, and it was analyzed that it was a beneficial class for non-major college students to feel the importance of the 4th industry.

IV. CONCLUSION

In this study, a CT-CPS model-based software liberal arts course was designed and applied to non-majors, and the effectiveness of creative problem-solving ability and learning motivation was verified. An independent sample t-test, paired sample t-test, and qualitative response analysis were performed for the analysis. The CT-CPS model-based software class had a positive effect on improving the creative problem-solving skills of non-majors. In particular, the experimental group showed a significant change in the 'Motivational elements' area among the sub-factors of the creative problem-solving ability test compared to the control group. In addition, as a result of comparing the pre-test and post-test for each test in the experimental group, there were statistically significant changes in all areas of 'Knowledge in a specific area, the function of thinking, skill, and mastery', 'Diffused thinking', 'Critical and logical thinking' and 'Motivational elements' of the creative problem-solving ability test.

The CT-CPS model-based software class had a positive effect on improving the learning motivation of non-majors. In particular, the experimental group showed a significant change in the 'Self-efficacy' area among the sub-factors of the learning motivation test compared to the control group. In addition, as a result of comparing the pre-post tests of the experimental group, among the sub-factors of the learning motivation test, there were significant changes in 'Internal goals', 'Control of learning beliefs', and 'Self-efficacy' excluding 'External goals'. Through qualitative analysis, it was confirmed that the software class to which the CT-CPS model was applied affected the cultivation of creative problem-solving ability and learning motivation through discovering and solving problems on their own in real life.

In summary, the CT-CPS model-based software class finds and recognizes problems to be solved in daily life or major, and it could be judged that it effectively improved learners' creative problem-solving ability and learning motivation in the process of imagining and solving ideas. It must be applied to various programming languages in the software class based on the CT-CPS model for non-majors. In this study, the app inventor programming language, which allows non-major college students to create android apps without professional programming knowledge easily, was used in the class. This study analyzed the CT-based app inventor class as a programming language suitable for non-majors to improve their basic understanding of coding. If the effectiveness of programming classes through other software can be verified, it will be a more reliable software class model.

Extracurricular activities for students who have taken software liberal arts classes for non-majors or activities linked to follow-up class programs should be prepared. It will be possible to further inspire learning motivation and a sense of achievement through software exhibitions, creative idea contests, and software experience activities, where the output of the class can be practically used.

There is a need to develop a learning model for cultivating the creative convergence capability of non-majors. The effectiveness of the CT-CPS model has already been confirmed in studies for cultivating computational thinking and creative problem-solving skills for elementary, middle, and high school students. However, it is time for systematic software education in connection with the university curriculums. Therefore, if the CT-CPS model is continuously modified and supplemented according to the software competency required by the times and a teaching and learning method is developed, software education can be made more effective.

Through this study, we hope it will be used as a basic background for university software liberal arts education, where the demand for software education is gradually increasing. In addition, we expect that various follow-up studies that suggest effective directions and approaches of software education for non-majors will be conducted.

REFERENCES

[1] Ministry of Education, "National Curriculum Revised 2015," Ministry of Education Notification, No. 2015-74, Sep. 2015.

[2] Ministry of Education, "A Support Plan for the Reinforcement of Software Education of the Teacher Training University in 2019," Ministry of Education Notification, No. 2019-84, Mar. 2019.

[3] J. M. Wing, "Computational thinking," Communications of the ACM, vol. 49, no. 3, pp. 33–35, Mar. 2006.

[4] W. S. Kim, "A Study on the Students' Perceptions Trend for Software Essentials Subject in University," *Korean Journal of General Education*. vol. 13, no. 4, pp. 161–180, Aug. 2019. to be published. DOI : 10.32431/kace.2015.18.3.005.

[5] I. H. Ahn, "Current status and future of coding education," *Media & Education*. vol. 6, no 1, pp.76—87, Jun. 2016.

[6] S. H. Kim, "Analysis of Non-Computer Majors' Difficulties in Computational Thinking Education," *The Journal of Korean Association of Computer Education*. vol. 18, no. 3, pp. 49–57, May. 2015. to be published. DOI: 10.32431/kace.2015.18.3.005.

[7] G. J. Park and Y. J. Choi, "Exploratory Study on the Direction of Software Education for the Non-major Undergraduate Students," *The Journal of Education & Culture*. vol. 24, no. 4, pp. 273–292, Aug. 2018, to be published. DOI: 10.24159/joec.2018.24.4.273.

[8] H. W. Kim, "A Study on the Design and Operation of Liberal Arts College for Improving Creativity of University Students," *The Journal of Creativity Education*. vol. 18, no. 4, pp. 91–114, Dec. 2018, to be published. DOI : 10.36358/JCE.2018.18.4.91.

[9] K. H. Lee, E. K. Kim and K. H. Lee, "The Relationships of Creativity, Problem Solving Ability and Motivation in College Students," *The Journal of Creativity Education*. vol. 3, no. 1, pp. 125–139, Mar. 2013.

[10] D. I. Park, "Relationships Between Elementary School Students' Academic Motivation, Metacognition, and Creative Problem-Solving Skills in Creative Problem Solving-Based On and Offline Integrated Learning," M.S. thesis, Dept. Edu. Eng., Ewha Womans Univ., Seoul, Korea, 2007.

[11] W. S. Han, J. K. Kim and H. K. Lee, "Effects of the 'Motivation Semester' Program among University Freshmen: Focused on K-University," *The Journal of the Korea Contents Association*. vol. 13, no. 7, pp. 522–530, Mar. 2013, to be published. DOI: 10.5392/JKCA.2013.13.07.522.

[12] E. I. Jeong and Y. H. Park, "Relations among faculty trust, learning motivation, and school adjustment of college students," *Asian Journal of Education*. vol. 9, no. 1, pp. 73–93, 2008, to be published. DOI : 10.15753/aje.2008.9.1.004.

[13] H. S. Lee, "Development of a Creativity Program for University Students through the Liberal Educational Approach," *Korean Journal of General Education*. vol. 11, no. 6, pp. 333–358, Dec. 2017.

[14] J. Y. Seo and S. H. Shin, "A Case Study on the Effectiveness of Major-friendly Contents in Software Education for the Non-majors," *The Journal of Digital Convergence*. vol. 18, no. 5, pp. 55–63, May. 2020, to be published. DOI: 10.14400/JDC.2020.18.5.055.

[15] J. E. Nah, “Software Education Needs Analysis in Liberal Arts,” *Korean Journal of General Education*. vol. 11, no. 3, pp. 63–89, Jun. 2017.

[16] S. W. Jung, H. S. Jang and S. J. Kim, “The Status and Implications of the Basic Liberal Arts Curriculum in Domestic Universities Applying the Standard Model of the Korea National Institute for General Education,” *Korean Journal of General Education*. vol. 14, no. 4, pp. 83–95, Oct. 2020.

[17] J. S. Sung and H. C. Kim, “Analysis on the International Comparison of Computer Education in Schools,” *The Journal of Korean Association of Computer Education*. vol. 18, no. 1, pp. 45–54, Jan. 2015, to be published. DOI: 10.32431/kace.2015.18.1.005.

[18] Harvard University. “CS50: Introduction to Computer Science,” Available: https://pll.harvard.edu/course/cs50-introduction-computer-science?delta=0, Accessed on: Feb. 10, 2022.

[19] Department for Education GOV.UK, “GCE AS and A level computer science,” Available: https://www.gov.uk/government/publications/gce-as-and-a-level-for-computer-science, Accessed on: Feb. 15, 2022.

[20] Oxford Cambridge and RSA, "Computer Science - H046, H446," Available: https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/, Accessed on: Feb. 15, 2022.

[21] Information Processing Society of Japan, “Curriculum Standard General Information Processing Education (GE),” Available: https://www.ipsj.or.jp/annai/committee/education/j07/ed_j17-GE.html, Accessed on: Feb. 20, 2022.

[22] M. J. Kim, J. M. Kim and W. G. Lee, “An Analysis of the Current Status of SW-Centered Universities' Informatics Curriculum in Korea based on Japan's Standards for Liberal Arts Informatics Curriculum,” *The Journal of Korean Association of Computer Education*. vol. 24, no. 4, pp. 13–23, Jul. 2021. to be published. DOI: 10.32431/kace.2021.24.4.002.

[23] W. S. Kim, “A Study on the Establishment of Essential Software Courses for Non-majors: Focusing on Soongsil University Computational Thinking Course Opening Process and Operational Evaluation,” *Proceedings of Korean Association of General Education Conf.,* 110–115, Nov. 2017.

[24] H. J. Jeong, “Analysis of Algorithm Thinking based on Basic Software Education Effectiveness for Non-major Students in SW college,” *Korean Journal of General Education*. vol. 15, no. 2, pp. 175–187, Apr. 2021.

[25] C. C. Shin, “The Effects of Computational Thinking-based Liberal Education on Problem Solving Ability,” *The Journal of the Korea Institute of Information and Communication Engineering*. vol. 25, no. 2, pp. 246–251, Feb. 2021.

[26] M. J. Kim, G. S. Yoo and H. C. Kim, “Development of a Scoring rubric based on Computational Thinking for Evaluating students` Computational Artifacts in Programming course,” *The Journal of Korean Association of Computer Education*. vol. 20, no. 2, pp. 1–11, Mar. 2017, to be published. DOI : 10.32431/kace.2017.20.2.001.

[27] Y. O. Song, “Design and Implementation of Reflection-based Coding Education: Case Study of ‘SW and Computational Thinking’ Courses at H University,” *The Journal of Educational Technology*. vol. 33, No. 3, pp.709–736, Sep. 2017. to be published. DOI : 10.17232/KSET.33.3.709.

[28] J. H. Jung, “A study on basic software education applying a step-by-step blinded programming practice,” *The Journal of Digital Convergence*. vol. 17, no. 3, pp. 25–33, Mar. 2019. to be published. DOI : 10.14400/JDC.2019.17.3.025.

[29] National Institute of the Korean Language, Standard Korean Dictionary, “Creativity,” Available: https://stdict.korean.go.kr/main/main.do, Accessed on: Feb. 25, 2022.

[30] J. P. Guilford, “The nature of human intelligence,” New York, NY, USA: McGrow-Hill, 1967.

[31] M. S. Basadur, Managing the creative process in organizations. In M. A. Runco, “Problem finding, problem solving, and creativity,” pp.237–268. Norwood, NJ, USA: Ablex, 1994.

[32] A. F, Osborn, Applied imagination, “principles and procedures of creative problem-solving,” 3rd revised edition, NY, USA: CEF, 1993

[33] S. H. Park, “Development and Application of Programming Education Model to Enhance Creative Problem Solving Abilities,” M. S. thesis, Dept. Element. Com. Edu., Seoul National Univ., Seoul, Korea, 2016.

[34] A. Y. Kim, “Self-efficacy and academic motivation,” *The Korean Journal of Educational Methodology Studies*. vol. 16, no. 2, pp. 1–38, Jan. 2004.

[35] J. J. Jeong, “A Factor Analysis of Motivation To Learn Among Korean Elementary School Children,” *Korean Journal of Psychological and Social Issues*. vol. 14, no. 1, pp. 167–186, Feb. 2008.

[36] K. H. Lee, E. K. Kim and K. H. Lew, “The Relationships of Creativity, Problem Solving Ability and Motivation in College Students,” *Korean Society for Creativity Education*. Vol. 3, No. 1, pp.125–139, Mar. 2013.

[37] Y. J. Jeon, “The Development and Application of a CT-CPS (Computational Thinking-based Creative Problem Solving) Instructional Model for the Software Education of New Curriculum,” Ph.D. dissertation, Dept. Info. Gift. Edu., Korea National Univ., Cheongju, Korea, 2017.

[38] Y. J. Jeon and T. Y. Kim, “The Analysis of Cognitive and Affective Effects on the CT-CPS Instructional Model for the Software Education Class in Middle School,” *The Journal of Korean Association of Computer Education*. vol. 20, no. 4, pp. 47-57, Jul. 2017, to be published. DOI : 10.32431/kace.2017.20.4.005.

[39] Y. J. Jeon and T. Y. Kim, “Analysis of Cognitive and Affective Effects in Elementary School Students` Software Class based on CT-CPS Model,” in *Proc. KACE*, Seoul, Korea, vol. 21, no. 1, pp25–28, Jan. 2017.

[40] J. H. Koo, Y. J. Jeon and T. Y. Kim, “The Development and Application of Lesson Contents Based on the CT-CPS Framework for Improving the Creative Problem Solving Ability of Elementary Informatics Gifted Students,” Korean Journal of Elementary Education. vol. 27, no. 2, pp. 339–357, Jun. 2016, to be published. DOI : 10.20972/kjee.27.2.201606.339.

[41] M. J. Kim, S. C. Lee and T. Y. Kim, “The Effect of Software Education Including Data Literacy on Computational Thinking and the Creative Problem-Solving Ability of Middle School Students,” *Korean Journal of Teacher Education*. vol. 37, no. 1, pp. 167–184, Jan. 2021, to be published. DOI : 10.14333/KJTE.2020.37.1.08.

[42] J. T. Kim and T. Y. Kim, “Development of SW Education program based on Living Lab process,” in *Proc. KACE*, Korea, Aug. 2021, vol. 254, no. 2, pp. 19-21.

[43] Y. R. Lee and T. Y. Kim, “CT-CPS based Informatics and Science Creative Convergence Class Design Using Physical Computing,” in *Proc. KACE*, Korea, Jan. 2020, vol. 24, no. 1, pp. 189–192.

[44] S. J. Hwang and Y. J. Lee, “Education using App Inventor on Informatics Gifted Elementary Students’ Creative Problem Solving Ability and Learning Flow,” M.S. thesis, Dept. Info. Gift. Edu., Korea National Univ., Cheongju, Korea, 2015.

[45] D. R. Kang and N. W. Kwon, “The Effects of Process and Result Oriented Motivation Approach on Motivation for Learning and Academic Achievements,” M.S. thesis, Dept. Element. Edu. Eng., Seoul National Univ., Cheongju, Korea, 1996.

[46] J. Hwang et al., “SW education entry & hamster that develops computational thinking ability,” Hanbit academy, Seoul, Korea, 2016, pp. 251–252.

[47] Best easy word cloud generator, “Create a word cloud”, Available: https://wordc.net/, Accessed on: Feb. 25, 2022.