

## 컴퓨팅사고력, 창의적 문제해결력 신장을 위한 대학 교양 SW 기초 교육의 효과 분석

노지예\*

덕성여자대학교 차미리사교양대학 조교수

### Analysis of the Effectiveness of Liberal SW Education focused on Developing Computational Thinking and Creative Problem Solving Ability

Jiyae Noh\*

College of Cha Mirisa Liberal Arts, Duksung Women's University

**요약** 대학 교양 SW 기초 교육은 SW를 기반으로 일상 생활의 문제를 창의적으로 해결할 수 있는 능력을 기르는 데 중점을 두어야 한다. 따라서, 본 연구에서는 컴퓨팅사고력, 창의적 문제해결력을 신장시키기 위한 수업을 설계하고, 수업의 효과를 검증하고자 하였다. 이를 위해 A 대학교 학생 38명을 대상으로 융합프로젝트, CT-CPS 모형을 고려하여 SW 기초 교육을 실시하고 그 효과를 분석하였으며, 연구 결과는 다음과 같다. 첫째, SW 기초 교육을 진행한 후 전체 학생들의 컴퓨팅사고력, 창의적 문제해결력이 유의하게 향상되었다. 둘째, 상위, 하위 집단의 컴퓨팅사고력과 하위 집단의 창의적 문제해결력이 유의하게 향상되었으나, 상위 집단의 창의적 문제해결력은 향상되지 않았다. 셋째, 인문사회계열과 자연과학계열의 컴퓨팅사고력과 자연과학계열의 창의적 문제해결력이 유의하게 향상되었으나, 인문사회계열의 창의적 문제해결력은 향상되지 않았다. 본 연구는, 컴퓨팅사고력, 창의적 문제해결력 신장에 초점을 두어 수업을 설계하고, 수업의 효과를 사전 검사 결과 및 계열에 따라 심층적으로 분석하였다는 점에서 연구의 의의를 찾을 수 있다.

**키워드** : 컴퓨팅사고력, 창의적 문제해결력, 사전 검사에 따른 수업 효과, 계열별 수업 효과, CT-CPS 모형

**Abstract** In liberal SW education, nurturing student with creative problem-solving ability based on SW is considered important. The purpose of this study is to design SW education and to investigate the effects on students' computational thinking and creative problem solving abilities. This study designed classes in accordance with convergent project and the CT-CPS model and 38 undergraduate students have participated this study. The questionnaire survey was given to students and analyzed the effectiveness of class. The results of this study were as follows: First, SW education significantly improved computational thinking and creative problem solving ability. Second, computational thinking improve significantly in high and low initial score group and creative problem solving improve significantly in low initial score group. However, creative problem solving ability did not improve significantly in high initial score group. Third, computational thinking improve significantly in all majors and creative problem solving improve significantly in college of natural science. However, creative problem solving ability did not improve significantly in college of humanities and social science. In examining the effects on students' computational thinking and creative problem-solving abilities and verify differences by pre-test and major, this study provides significance in expanding the understanding about the nature liberal SW education.

**Key Words** : Computational thinking, Creative problem solving ability, Effectiveness by pre-test, Effectiveness by major, CT-CPS model

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea(NRF-2019S1A5A8036420)

\*Corresponding Author : Jiyae Noh(jynoh@duksung.ac.kr)

Received October 15, 2022

Accepted January 20, 2023

Revised November 30, 2022

Published January 28, 2023

## 1. 서론

미래 사회를 살아갈 학생들은 타 분야와의 융합을 기반으로 창의적으로 문제를 해결하는 능력이 중요하므로 [1], 교육 현장에서 SW 교육에 대한 관심이 높아지고 있다. 초, 중등학교에서 SW 교육이 의무화되면서, SW 교육은 양적으로 크게 확대되었으며, 학생 및 교사의 SW 교육의 중요성에 대한 인식이 높아지고 있다.

대학에서도 교양 SW 교육이 빠른 속도로 증가하고 있으며, 기초 SW 교과목을 교양 필수 과목으로 전환하는 학교 또한 증가하는 추세이다. 하지만 대학 SW 수업은, 여전히 실습이나 기능 위주로 진행되거나, 교재의 실습 예제를 따라하는 식으로 진행되는 경우가 있으며[2], 어떤 방법으로 교육을 진행해야 하는지에 대해 구체적인 자료가 많지 않은 실정이다.

다가올 지능정보화 사회에서는 SW를 통해 자신의 생각을 표현하고, 컴퓨팅사고력을 기반으로 일상생활의 문제를 창의적으로 해결할 수 있는 역량이 필요하므로[2,3], 컴퓨팅사고력과 창의적 문제해결력 등 고등사고력에 중점을 두어 수업을 설계하는 것이 중요하다. 본 수업에서는 학생들의 컴퓨팅사고력과 창의적 문제해결력을 신장시키기 위하여, 학생들이 스스로 융합프로젝트를 완성하도록 하였으며, CT-CPS 모형을 추가적으로 고려하여 수업을 설계하고자 한다. 이 모형은 컴퓨팅사고력(CT: Computational Thinking) 요소와 창의적 문제해결(CPS: Creative Problem Solving) 단계와의 상호 관련성을 분석한 모형으로[4], 대학 SW 교육에 적용하기에 적합하므로, 대학 SW 교육에 시사점을 줄 수 있다.

한편, 기초 SW 교육에서 학생들의 높은 체감 난이도는 SW 교육에서 여전히 중요한 이슈가 되고 있다[5]. 대학에서의 SW 교육은 대부분 프로그래밍을 처음 접하는 학생들을 대상으로 진행되므로, 프로그래밍 수업에서 학생들은 지속적으로 어려움을 호소하고 있으며, 중도탈락 또한 많이 발생한다[6-8]. 따라서, SW 교육에서 수업 전 학습자의 수준을 진단하고, 학습자의 어려움에 대한 분석이 고려되어야 한다.

더불어, 대학 SW 교육 환경에서 학습자의 전공 연계성 여부에 관심을 가져야 하며, 더 나아가 계열별, 전공별 연계성을 높일 수 있는 기초 SW 교육과정의 개발이 필요하므로[9]. 이를 위해 계열별 학습자 분석이 선행되어야 할 필요가 있다. 하지만, 계열별 학습성과 차이에 관한 연구들[9-13]은 연구 결과가 일관되지 않으며, 변인에 따라

차이를 보이는 연구들도 있으므로[12], 추가적인 연구가 필요하다.

따라서 본 연구는 학생들의 컴퓨팅사고력, 창의적 문제해결력을 신장시키는 데 중점을 두어 수업을 설계하고, 학생들의 사전 검사 결과 및 계열에 따른 수업의 효과를 분석하고자 하였다. 본 연구의 구체적인 연구문제는 다음과 같다.

연구문제 1. 컴퓨팅사고력, 창의적 문제해결력 신장을 위한 SW 수업은 학생들(전체, 집단별, 계열별)의 컴퓨팅사고력을 향상시키는가?

연구문제 2. 컴퓨팅사고력, 창의적 문제해결력 신장을 위한 SW 수업은 학생들(전체, 집단별, 계열별)의 창의적 문제해결력을 향상시키는가?

## 2. 선행연구

### 2.1 CT-CPS 모형

CT-CPS 모형은 SW와 관련된 사고과정과 원리를 실생활과 관련지어 창의적으로 해결 방안을 구현하는 수업 구성 원리로, 문제 인식 및 분석, 아이디어 구상, 설계, 구현 및 평가의 4단계로 구성되어 있다[4].

CT-CPS 모형은 주로 초, 중등교육 맥락에서 SW 교육의 효과를 검증하는 연구에서 사용되었다. 전용주, 김태영(2017)[14]의 연구에서는 중학생들을 대상으로 CT-CPS 모형을 적용하여 수업을 실시하고, 사전, 사후 검사 결과를 비교하여 그 효과를 분석하였다. 연구 결과, 인지적 측면과 정의적 측면 모두(창의적 문제해결력, 메타인지, 학습동기)에서 학생들이 긍정적으로 변화되었음을 밝혔다.

초등학생을 대상으로 한 연구에서도 유사한 결과가 보고되었다. 전용주, 김태영(2017)[15]은 초등학생을 대상으로 CT-CPS 모형을 적용하여 수업을 실시하고, 그 효과를 분석하였다. 연구 결과, 창의적 문제해결력, 메타인지, 컴퓨터 학습 태도, 학습동기의 대부분의 하위요인에서 통계적으로 유의한 향상이 관찰되었으며, CT-CPS 모형에서 문제 및 해결 과정을 스스로 찾는 과정이 인지적, 정의적 역량에 도움이 되었다고 밝혔다.

최근 대학 교양 SW 교육에서도 CT-CPS 모형을 적용한 연구가 보고되었다. 박희정, 전용주(2021)[16]의 연구에서는, CT-CPS 모형을 기반으로 비전공자 대상 교양 SW 수업을 설계하였다. 수업은 중간고사 이전까지 문법 학습 이후 실생활의 예제를 다루며, 중간고사 이후에 CT-CPS 모형을 활용한 미니프로젝트를 수행하도록 구

성되었다. 박희정, 전용주(2021)[16]는 SW 교육이 확산되고 있으나, 대학생을 위한 SW 교육 프로그램이 부족하므로, 다양한 방법이 모색되어야 한다고 언급하였다.

CT-CPS 모형은 인지적 영역 뿐 아니라, 정의적 영역에도 효과적이며[14,15], 컴퓨팅사고력과 창의적 문제해결 단계를 종합적으로 고려하였으므로[4], 대학 SW 교육에 적용하기에 적합하다. 따라서, 본 연구에서는 CT-CPS 모형을 고려하여 수업을 설계하고 그 효과를 검증하고자 한다.

## 2.2 컴퓨팅사고력

컴퓨팅사고력은 추상화와 자동화에 기반한 문제 해결 능력으로[17], 해결해야 할 문제를 접했을 때 컴퓨터과학 자처럼 사고하는 것을 의미한다[17,18]. 대학 교양교육에서 기초 SW 교육의 중요성이 강조되면서, 상당수의 대학에서 비전공 학생을 포함한 모든 학생에게 SW 교육을 실시하는 추세이며[19], 컴퓨팅사고력은 전공자 뿐 아니라 모든 사람들이 갖추어야 할 역량이라는 인식 또한 확산되고 있다.

김민자, 김현철(2018)[10]의 연구에서는 핵심 교양 컴퓨팅 강좌를 수강한 비전공 대학생 505명을 대상으로, 컴퓨팅사고력 기반 역량 평가 문항으로 사전-사후 검사를 실시하였다. 연구 결과, 수강생들의 컴퓨팅사고력 점수가 통계적으로 유의하게 향상되었으며, 수강생들의 계열별(컴퓨터 전공, 이공계열, 인문사회계열) 컴퓨팅사고력 차이는 통계적으로 유의하지 않았다.

교육대학교 학생들을 대상으로 한 연구에서도 유사한 연구 결과가 보고되었다. 박선주(2021)[20]의 연구에서는, 교육대학교 1학년 학생 35명을 대상으로 교양 필수 과목인 SW·AI 교육을 실시하였다. 연구 결과, 학생들의 컴퓨팅사고력(분해, 패턴인식, 추상화, 알고리즘), SW 문해력이 통계적으로 유의하게 향상되었다. 또한, 박선주(2021)[20]는 SW·AI 교육은 학생들이 컴퓨팅사고력 및 SW 교육의 중요성을 깨닫고, 문제 해결 과정을 표현, 이해하는 데 도움을 줄 수 있다고 하였다.

SW 교육 프로그램을 설계하고, 컴퓨팅사고력이 신장되었는지 검증한 연구도 보고되었다. 이재호, 이승훈(2021)[21]의 연구에서는 267명의 교육 소외계층 학생들을 대상으로, 동영상 강의와 실시간 강의를 혼합한 SW 교육 프로그램(비긴즈 프로그램, 언플러그드 프로그램, 게임메이커 프로그램)을 적용하였다. 이 프로그램은 직접교

수모형을 기반으로 설계되었으며, 학생들이 미션을 해결하는 활동을 진행하도록 구성되었다. 연구 결과, 수업에 참여한 학생들은 컴퓨팅사고력의 모든 하위요인(추상화, 문제분해, 알고리즘, 자동화, 데이터처리) 점수가 통계적으로 유의하게 향상되었다.

선행연구에 따르면, SW 교육은 학생들의 컴퓨팅사고력 향상에 긍정적임을 확인할 수 있다[10][20,21]. 컴퓨팅사고력은 읽기, 쓰기, 셈하기처럼 기본적으로 갖추어야 할 역량이며[17,18], 대학생들이 전공에 상관없이 누구나 갖추어야 할 능력이므로, 대학 기초 SW 교육에서 우선적으로 고려되어야 한다.

## 2.3 창의적 문제해결력

창의적 문제해결이란, 단순히 기존의 정보를 적용하여 문제를 해결하지 않고, 독창적인 아이디어로 문제를 해결하는 것을 의미한다[22]. 창의적 문제해결력은 확산적 사고, 수렴적 사고, 지식, 동기적 요인 등 여러 요소들의 상호작용으로 나타난다는 특성이 있다[22].

선행연구에 따르면, SW 교육은 학생들의 창의적 문제해결력에 긍정적인 영향을 미친다. 김경태, 이철현(2021)[23]의 연구에서는 SW 교육 경험이 없는 초등학교 고학년 학생 56명을 실험집단(마이크로비트 기반 SW 교육)과 통제집단(엔트리, 햄스터 로봇을 활용 SW 교육)으로 나누어 SW 교육을 실시하였다. 연구 결과, 실험집단 학생들의 창의적 문제해결력이 통계적으로 유의하게 향상되었음을 밝혔다.

교수설계모형을 고려하여 SW 교육을 실시하고, 창의적 문제해결력이 신장되었는지 검증한 연구도 있다. 전용주, 김태영(2017)[14]은 중학생들을 대상으로 CT-CPS 모형을 기반으로 한 SW 교육을 실시하였다. 수업은 정보 교과 및 실생활과 관련된 내용을 포함하고 있으며, 인지적, 정의적 역량을 균형있게 함양할 수 있도록 구성되어 있다. 연구 결과, 학생들의 창의적 문제해결력 및 하위요인 일부(확산적 사고, 논리비판적 사고, 동기적 요소)가 통계적으로 유의하게 향상되었다.

대학생을 대상으로 한 연구에서도 유사한 결과가 보고되었다. 서응교, 신윤희(2020)[24]의 연구에서는 비IT전공 학생들을 대상으로, 창의적 사고 및 코딩 교육에 중점을 둔 디자인씽킹 기반 코딩교육 프로그램을 적용하였다. 연구 결과, 학생들의 문제해결력과 창의적 잠재력이 통계적으로 유의하게 향상되었음을 밝혔다.

선행연구에 따르면, SW 교육은 학생들의 창의적 문제 해결력 향상에 긍정적인 영향을 확인할 수 있다[14][23,24]. 학생들은 자신의 생각을 창의적으로 표현하고, 전공 영역의 문제를 융합적 관점에서 창의적으로 해결할 수 있는 역량을 기르는 것이 중요하므로[25], 창의적 문제해결력은 SW 교육에서 중요한 의미를 가진다.

#### 2.4 사전 검사 결과에 따른 차이

대학 교양 SW 기초 교육에 참여하는 학생들은, 대부분 프로그래밍 경험이 많지 않아 수업을 어렵게 느끼는 경우가 많으므로, SW 교육에서 학생들의 수준이 중요하게 고려되어야 한다. Moreno-Leon, Robles와 Roman-Gonzalez (2015)[26]은 학생들을 사전 검사 결과로 구분하여 수업의 효과를 심층적으로 분석하였다. 구체적으로, 88명의 스페인 10-14세 학생들을 대상으로 스크래치 교육을 실시하였으며, 사전 검사 결과에 따라 전체 학생을 상, 중, 하 집단으로 구분하였다. 연구 결과, 중, 하 집단 학생들의 컴퓨팅사고력은 통계적으로 유의하게 향상되었으며, 특히 중 집단 학생들의 컴퓨팅사고 점수가 가장 큰 향상을 보였다. 하지만, 상 집단 학생들의 컴퓨팅사고력은 통계적으로 유의하게 향상되지 않았다.

국내 대학생을 대상으로, 사전 검사 결과에 따른 학습 성과의 차이를 검증한 연구도 보고되었다. 박주연(2021)[12]의 연구에서는 대학 신입생 132명을 대상으로 교양 컴퓨팅사고 수업을 실시하였다. 이 연구에서는, 사전 컴퓨팅효능감과 사전 컴퓨팅흥미에 따라 사후 검사에 차이가 있는지 분석하기 위하여, 평균을 기준으로 전체 학생을 상위 그룹과 하위 그룹으로 구분하였다. 연구 결과, 사전 컴퓨팅효능감과 사전 컴퓨팅흥미 상위 집단과 하위 집단은 사후 검사의 모든 요인(융합적 사고, 융합적 가치 창출, 컴퓨팅효능감, 컴퓨팅흥미)에서 통계적으로 유의한 차이를 보였다.

선행연구에서는 사전 검사 결과를 기반으로 집단을 구분하여 수업의 효과를 분석하였으나, 연구 결과가 일관되게 보고되고 있지 않으므로, 추가적인 연구가 필요하다. 또한, 프로그래밍 수업은 학생들이 어렵게 느끼며, 중도 탈락이 많이 발생하므로[6-8], 수업 전 학습자의 수준을 진단하고, 이를 기반으로 수업의 효과를 분석하는 것이 필요하다.

#### 2.5 계열에 따른 차이

SW 교육에서 계열별, 전공별 차이에 관한 연구들이 진

행되고 있으나, 연구 결과가 일관되게 보고되고 있지 않다. 오미자, 김미량(2018)[11]의 연구에서는, 서울 소재 대학의 신입생 308명을 대상으로 프로그래밍 교육을 진행하였으며, 계열(자연공학계열, 인문사회계열, 예체능계열)에 따른 차이를 알아보았다. 연구 결과, 자연공학계열 학생들은 인문사회계열, 예체능계열 학생들에 비해 흥미 향상, 사고력향상, 성취도향상이 높게 나타났으며, 이는 통계적으로 유의하였다. 하지만, 인문사회계열과 예체능계열의 차이는 통계적으로 유의하지 않았다.

SW 교육에 대한 대학생들의 인식이 계열별로 차이가 있다고 밝힌 연구도 있다. 권정인(2019)[9]의 연구에서는, 321명의 대학 신입생들을 대상으로, SW 교육에 대한 인식의 계열별 차이를 탐색하기 위하여, 컴퓨팅사고력의 요소 중 문제해결 과정에서 중요하다고 생각하는 요소와, 소프트웨어개발 과정에서 중요하다고 생각하는 요소에 관해 설문을 진행하였다. 연구 결과, 자연계열 학생들은 문제해결 과정과 소프트웨어 개발 과정 모두 자료분석, 알고리즘, 자동화의 요소를 중요하다고 인식하였다. 하지만 인문계열 학생들은 문제해결 과정에서 자료수집, 문제분해, 자동화를 중요하다고 인식하였으며, 소프트웨어 개발 과정에서는 자료수집, 알고리즘, 자동화의 요소를 중요하다고 인식하여 계열별로 차이를 보였다.

한편, SW 교육 성취도에 계열별 차이가 나타나지 않았다고 보고한 연구도 있다. 서주영(2017)[13]의 연구에서는, 동일한 프로그래밍 수업을 수강한 SW 전공 학생 45명, SW 비전공(인문계열, 이공계열) 학생 21명을 대상으로 성취도(실습, 과제, 팀프로젝트, 시험, 출석을 포함한 종합 성적) 차이를 분석하였다. 연구 결과, SW 전공, 비전공 학생의 성취도 차이는 통계적으로 유의하지 않았으며, 비전공 학생들인 인문계열, 이공계열의 차이도 통계적으로 유의하지 않았다.

사후 검사에서는 계열별 차이가 통계적으로 유의하지 않았으나, 사전 검사에서는 계열별 차이가 있음을 보고한 연구도 있다. 박주연(2021)[12]의 연구에서는 대학 신입생 132명(인문사회대학, 과학기술대학)을 대상으로 융합적 사고력 신장을 위한 컴퓨팅사고 수업을 실시하였다. 연구 결과, 사후 검사에서는 모든 변인(융합적 사고, 융합적 가치창출, 컴퓨팅 효능감, 컴퓨팅 흥미)의 계열별 차이가 통계적으로 유의하지 않았다. 하지만, 사전 검사에서는 과학기술대학의 컴퓨팅효능감, 컴퓨팅 흥미의 평균 점수가 인문사회대학 학생들의 평균보다 높았으며, 그 차이가 통계적으로 유의하였다.

SW 교육의 효과나 인식에 계열별, 전공별 차이가 있다고 밝힌 선행연구[9][11]도 있으나, 계열별 차이가 없다고 보고한 연구[13]와 사전, 사후 검사에 따라 결과가 달라진다고 보고한 연구[12]도 있는 등 연구 결과에 차이가 있으므로, 추가적인 연구가 필요하다.

### 3. 연구방법

#### 3.1 연구대상 및 연구절차

본 연구에서는 2021년도 2학기(9월 1일~12월 21일)에 A 대학교 기초 SW 교과목을 수강한 학생들을 대상으로 수업을 진행하였다. 학생들은 대부분 2021년 1학기에 입학한 1학년 학생들로, 인문사회계열, 자연과학계열 학생들로 구성되어 있었으며, 상당수의 학생들이 SW 교육 경험이 없는 초급 수준의 학습자들이었다.

수업은 15주 동안 진행되었으며, 실습이 시작되는 9주차에 사전 설문을 실시하였고, 실습이 끝나는 15주차에 사후 설문을 실시하였다. 사전·사후 설문 중 하나라도 응답하지 않거나, 불성실하게 응답한 학생을 제외한 38명(인문사회계열 19명, 자연과학계열 19명)을 최종 연구대상으로 선정하였다.

#### 3.2 SW 교육을 위한 교수-학습 설계

본 연구는 A 대학교 기초 SW 교과목 수강생을 대상으로 진행되었으며, 컴퓨팅사고력을 함양하고, 이를 전공 영역 및 관심 분야와 융합하여 창의적으로 문제를 해결하는 것을 수업의 목표로 설정하였다.

수업 전반부(1주차-8주차)에는 컴퓨팅사고력 관련 이론 학습을 진행하였다. 구체적으로, 미래 사회의 변화와 SW의 중요성, 컴퓨팅사고력의 개념과 구성요소 등을 사례 기반으로 학습하였으며, 컴퓨팅사고력을 기반으로 전공 및 일상 생활의 문제에 접근하는 토론 활동을 실시하였다.

수업 후반부(9주차-15주차)에는 스크래치를 활용한 실습을 진행하였으며, 실습을 통해 순차, 선택, 반복, 변수, 함수, 자료구조, 인공지능 등에 대해 학습하였다. 또한, 실생활과 밀접한 예제를 스크래치를 기반으로 해결하는 활동을 진행하였는데, 학생들이 스스로 본인 전공 및 일상 생활과 관련된 문제를 찾고, 이와 관련된 융합프로젝트 계획서를 작성하도록 하였다. 융합프로젝트 계획서는 교수자의 피드백을 기반으로 수정하는 과정을 거쳤으며, 수정 내용을 토대로 학기말에 스크래치 프로젝트를

과제로 제출하도록 하였다. 주차별 수업 내용과 수업 방법은 Table 1과 같다.

Table 1. Lesson plan and educational methods

Week	Lesson plan	Educational Methods
1	• Orientation • Importance of SW	• Explain • Lecture • Debate • Discussion
2	• Importance of computational thinking	
3	• Decomposition	
4	• Pattern recognition	
5	• Abstraction	
6	• Algorithm	
7	• Algorithm • Automation	
8	• Mid-term exam	
9	• Sequence • Selection • Convergence project plan	• Debate • Practice • Project based learning
10	• Iteration	
11	• Variable	
12	• Function • Convergence project plan feedback	
13	• Artificial intelligence	
14		
15	• Convergence project	

본 연구에서는 학생들의 컴퓨팅사고력, 창의적 문제해결력을 신장시키기 위하여, 융합프로젝트와 함께 CT-CPS 모형을 추가적으로 고려하였다. CT-CPS 모형은 컴퓨팅사고력의 요소와 CPS(Creative Problem Solving)

Table 2. Example of lesson (CT-CPS model)

Steps of CT-CPS model		Examples of lesson
Problem identification and analysis	Motivation and SW selection	• Motivating students by explaining the importance of problem solving • Thinking about the connection between CT and majors used to solve problems • Introduction about EPL (Educational programming language) and Scratch
	Problem Recognition and Problem Definition	• Finding real-life problems related to computational thinking • Describing why you want to solve this problem • Expressing the final problem to be solved
Conception	Thinking of various ideas	• Find the core of the problem • Thinking of various ideas that can solve problems through SW • Organize ideas through brainstorming, mind mapping and SCAMPER
	Simplifying ideas	• Simplifying ideas (Abstraction)
Design	Designing ideas	• Expressing ideas step by step and creating flowcharts (Algorithm)
Implementation	Implementing ideas	• Implementing ideas using Scratch • Debugging

단계를 종합적으로 고려한 모형으로, 문제 인식 및 분석, 아이디어 구상, 설계, 구현 및 평가 단계로 구성된다 [4][16]. 이 모형은 문제 인식 및 분석 단계에서 스스로 문제를 찾고 이를 컴퓨팅사고력을 통해 해결한다는 특징이 있으며[4], SW 교육 관련 선행연구에서 활용된 바 있다 [4][14][16][23][27,28]. CT-CPS 모형을 기반으로 한 단계별 수업 설계 예시는 Table 2와 같다.

문제 인식 및 분석 단계에서는 SW에 대한 동기를 유발하고, 문제를 정의하도록 하였다. 구체적으로, 학생들은 나의 전공 영역에서 일어날 변화와 관련된 동영상을 찾아 공유하거나, 전공 분야에서 일어날 변화는 무엇인지 자신의 의견을 작성하는 활동을 진행하였다. 또한, 생활 속에서 컴퓨팅사고력을 활용한 예시(문제와 해결 방법)를 이야기 해 보고, 자신이 해결하고자 하는 문제를 정의(현재 상태, 목표 상태)하는 활동을 진행하였다. 문제 인식 및 분석 단계의 학생 활동 예시는 Table 3과 같다.

Table 3. Example of students' activity(problem identification and analysis)

Students' major	Examples of Students' activity
College of humanities and social science	"Kindergarten facilities seem to become a little safer through the IoT. For example, it would be useful to have a system that automatically turns off the fire when you need to move quickly while cooking meals." "In the field of early childhood education, there is an increasing demand from parents who want to teach coding from an early age."
College of natural science	"In the field of IT-related majors, many unimaginable technologies are likely to arise in the future. I think it is necessary to develop the ability to generate new ideas and computational thinking in order to respond to these changes." "Due to the 4th industrial revolution, the IoT or big data is used to control the dosage of medicine in real time and to speed up the development of new medicine." "In particular, in the case of bio engineering, numerous experimental data are particularly important, and data collection, analysis, and statistics have become easier with software technology and the IoT. In order to respond to these changes, it is necessary to develop the capability to apply the analysis results to other fields."

아이디어 구상 단계에서는 아이디어를 산출하고, 아이디어를 정리, 단순화하도록 하였다. 구체적으로, 일상 생활의 문제를 SW 기반으로 해결하기 위한 아이디어를 마인드맵, 브레인스토밍, SCAMPER 등을 통해 다양하게 떠올려 보고, 자신의 아이디어를 정리하여 단순화하는 추상화 활동을 진행하였다. 아이디어 구상 단계의 학생 활동 예시는 Fig. 1과 같다.

설계 단계에서는 단순화한 아이디어를 알고리즘으로 표현하도록 하였다. 구체적으로, 자연어, 의사코드, 순서

도 등을 통해 아이디어를 순서대로 표현하는 활동을 진행하였다. 설계 단계에서의 학생 활동 예시는 Fig. 2와 같다.



Fig. 1. Decomposition of Jeju Island travel (conception)

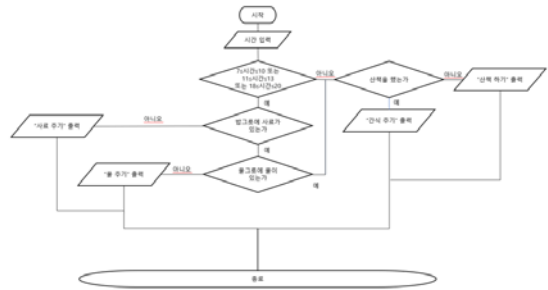


Fig. 2. Why dogs bark? (Design)

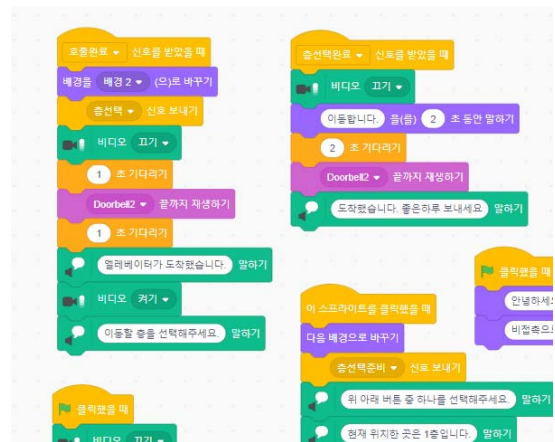


Fig. 3. Untact elevator call program (implementation)

구현 및 평가 단계에서는 자신의 아이디어를 구현하고, 평가, 수정하도록 하였다. 구체적으로, 학생들은 자신

의 아이디어를 스케치로 구현, 디버깅하는 활동을 진행하였다. 구현 및 평가 단계에서의 학생 활동 예시는 Fig. 3과 같다.

### 3.3 측정도구

#### 3.3.1 컴퓨팅사고력

컴퓨팅사고력을 측정하기 위하여, 최형신(2014)[29]의 도구를 본 연구 상황에 맞게 수정하여 사용하였다. 이 도구는 Likert 5점 척도의 14문항으로, 본 연구에서의 Cronbach's  $\alpha$ 는 .95이다.

이 도구는 개념, 수행, 관점의 3개의 하위요인으로 구성되어 있으며, 개념 7문항(예: 나는 시작 버튼을 클릭하면 스프라이트가 움직이도록 할 수 있다), 수행 4문항(예: 나는 만약 스프라이트가 잘못된 길로 갈 경우 그것을 인식하고 수정할 수 있다), 관점 3문항(예: 나는 스프라이트를 원하는 대로 움직이도록 프로그래밍 할 수 있다)으로 구성되어 있다.

또한 이 도구는 사전-사후 검사의 형태로 구성되어 있어 SW 교육의 효과를 측정하는 것이 가능하며, SW 교육 관련 선행연구[29,30]에서 컴퓨팅사고력 측정 도구로 사용된 바 있다.

#### 3.3.2 창의적 문제해결력

창의적 문제해결력을 측정하기 위하여, 서울대학교 심리 연구실 MI 연구팀이 한국교육개발원[31]의 연구를 기반으로 개발한 도구를 정은영(2008)[32], 전용주(2017)[4]의 연구에서 활용한 것을 사용하였다. 이 도구는 Likert 5점 척도의 20문항으로, 본 연구에서의 Cronbach's  $\alpha$ 는 .96이다.

이 도구는 특정영역의 지식/사고기능/기술의 이해 및 숙달 여부, 확산적 사고, 비판적/논리적 사고, 동기적 요소의 4가지 하위 요인으로 구성되어 있으며, 특정영역의 지식/사고기능/기술의 이해 및 숙달 여부 5문항(예: 나는 많은 일에 호기심을 가지고 계속 질문한다), 확산적 사고 5문항(예: 나는 문제를 풀어낼 아이디어를 다양하고 풍부하게 만들어 낸다), 비판적/논리적 사고 5문항(예: 나는 실제로 있는 사실과 상상을 구별할 줄 안다), 동기적 요소 5문항(예: 나는 이 과목의 다른 주제에 대해서도 더 알고 싶다)으로 구성되어 있다.

이 도구는 SW 교육에서 창의적 문제해결력을 측정하는 데 적합하여 SW 교육 관련 선행연구[4][14][23]에서 측정도구로 사용되었으며, 본 연구에서 활용한 CT-CPS

모형 관련 연구[4][14,15][23]의 측정 도구로 사용된 바 있다.

### 3.4 자료 분석 방법

본 연구에서는 SPSS를 사용하여 수집된 자료에 관한 분석을 실시하였다. 본 수업이 컴퓨팅사고력, 창의적 문제해결력에 미치는 효과를 알아보기 위해 대응 표본  $t$ 검정을 실시하였다. 또한 본 수업의 효과에 관해 보다 심층적으로 분석하기 위하여, 전체 학생을 사전 검사 점수(상위 집단, 하위 집단) 및 계열(인문사회계열, 자연과학계열)로 나누어 대응 표본  $t$ 검정을 실시하였다.

## 4. 연구결과

### 4.1 전체 학생들의 수업 효과

#### 4.1.1 컴퓨팅사고력

수업의 효과를 분석하기 위하여, 컴퓨팅사고력 사전 검사와 사후 검사의 차이에 대한 대응 표본  $t$ 검정을 실시하였다. 그 결과, 학생들의 컴퓨팅사고력 점수의 차이는 통계적으로 유의한 것으로 나타났다( $t = -6.383, p = .000$ ).

컴퓨팅사고력 하위요인에 대한 대응 표본  $t$ 검정을 실시한 결과, 개념( $t = -6.886, p = .000$ ), 수행( $t = -5.757, p = .000$ ), 관점( $t = -5.814, p = .000$ )의 평균은 모두 통계적으로 유의하게 향상된 것으로 나타났다.

Table 4. The effect of education (computational thinking)

Variable / Sub-factor	pre		post		$t$	$p$
	$M$	$SD$	$M$	$SD$		
Computational Thinking	3.086	.797	4.094	.719	-6.383*	.000
Computational concepts	3.031	.784	4.135	.754	-6.886*	.000
Computational practices	3.099	.847	4.053	.736	-5.757*	.000
Computational perspectives	3.184	.862	4.053	.721	-5.814*	.000

\* $p < .05$

#### 4.1.2 창의적 문제해결력

수업의 효과를 분석하기 위하여, 창의적 문제해결력 사전 검사와 사후 검사의 차이에 대한 대응 표본  $t$ 검정을 실시하였다. 그 결과, 학생들의 창의적 문제해결력 점수의 차이는 통계적으로 유의한 것으로 나타났다( $t = -4.082, p = .000$ ).

창의적 문제해결력 하위요인에 대한 대응 표본  $t$ 검정

을 실시한 결과, 특정영역의 지식/사고기능/기술의 이해 및 숙달 여부( $t = -5.885, p = .000$ ), 확산적 사고( $t = -5.363, p = .000$ ), 동기적 요소( $t = -2.268, p = .029$ )의 평균은 유의하게 증가한 것으로 나타났다. 비판적/논리적 사고( $t = -1.279, p = .209$ )는 수업을 실시한 이후 평균 점수가 향상되었으나, 그 차이가 통계적으로 유의하지 않았다.

**Table 5. The effect of education (creative problem-solving ability)**

Variable / Sub-factor	pre		post		t	p
	M	SD	M	SD		
Creative problem-solving ability	3.428	.513	3.767	.660	-4.082*	.000
Understanding and attainment of skill /ability in specific areas	2.958	.656	3.395	.718	-5.885*	.000
Divergent thinking	3.121	.782	3.621	.720	-5.363*	.000
Critical /logical thinking	3.868	.583	4.021	.754	-1.279	.209
Motivational factors	3.763	.611	4.032	.772	-2.268*	.029

\* $p < .05$

## 4.2 집단별 수업 효과

### 4.2.1 컴퓨팅사고력

사전, 사후 컴퓨팅사고력 점수의 차이에 관해 보다 심층적으로 분석하기 위하여, 사전 컴퓨팅사고력 점수에 따라 전체 38명의 학생을 학생을 상위 집단 19명(사전 컴퓨팅사고력 점수 3.071~5.000인 집단)과 하위 집단 19명(사전 컴퓨팅사고력 점수 1.000~3.000인 집단)으로 구분하여 대응 표본  $t$  검정을 실시하였다. 그 결과, 상위 집단 ( $t = -2.417, p = .026$ )과 하위 집단( $t = -8.611, p = .000$ ) 모두 학생들의 컴퓨팅사고력 평균 점수가 향상되었으며, 사전-사후 점수의 차이는 통계적으로 유의하였다.

하위요인별로 대응표본  $t$  검정을 실시한 결과, 하위 집단에서는 모든 하위요인에서 통계적으로 유의한 향상을 보였으나, 상위 집단에서는 일부 하위요인에서만 유의한 향상을 보였다.

구체적으로, 하위 집단에서는 개념( $t = -8.051, p = .000$ ), 수행( $t = -8.050, p = .000$ ), 관점( $t = -8.172, p = .000$ )의 평균 점수가 모두 향상되었고, 그 차이가 통계적으로 유의하였다. 반면, 상위 집단에서는 개념( $t = -3.054, p = .007$ )의 평균 점수가 향상되었고, 그 차이는 통계적으로 유의하였으며, 수행( $t = -1.922, p = .071$ ), 관점( $t = -1.472, p = .158$ )은 평균 점수가 향상되었으나, 그 차이가 통계적으로 유의하지 않았다.

**Table 6. The effect of education by group (computational thinking)**

	Sub-factor	M(SD)		t	p
		pre	post		
High initial score (n=19)	Computational Thinking	3.684 (.495)	4.188 (.878)	-2.417*	.026
	Computational concepts	3.579 (.559)	4.226 (.872)	-3.054*	.007
	Computational practices	3.724 (.526)	4.132 (.910)	-1.922	.071
	Computational perspectives	3.825 (.560)	4.175 (.905)	-1.472	.158
Low initial score (n=19)	Computational Thinking	2.489 (.555)	4.000 (.523)	-8.611*	.000
	Computational concepts	2.482 (.561)	4.045 (.626)	-8.051*	.000
	Computational practices	2.474 (.612)	3.974 (.520)	-8.050*	.000
	Computational perspectives	2.544 (.590)	3.930 (.466)	-8.172*	.000

\* $p < .05$

### 4.2.2 창의적 문제해결력

사전, 사후 창의적 문제해결력 점수의 차이에 관해 보다 심층적으로 분석하기 위하여, 창의적 문제해결력 점수에 따라, 전체 38명의 학생을 동점자를 고려하여 상위 집단 18명(사전 창의적 문제해결력 점수 3.40-5.00)과 하위 집단 20명(사전 창의적 문제해결력 점수 2.65-3.35)으로 구분하여 대응 표본  $t$  검정을 실시하였다. 그 결과, 하위 집단 학생들의 창의적 문제해결력 점수가 향상되었으며, 사전-사후 점수의 차이는 통계적으로 유의하였다( $t = -5.269, p = .000$ ). 반면, 상위 집단 학생들의 창의적 문제해결력 평균 점수가 향상되었으나, 사전-사후 점수의 차이는 통계적으로 유의하지 않았다( $t = -1.687, p = .110$ ).

하위요인별로 대응표본  $t$  검정을 실시한 결과, 하위 집단에서는 4개 요인 모두 통계적으로 유의한 향상을 보였으나, 상위 집단에서는 일부 하위요인에서만 유의한 향상을 보였다.

구체적으로, 하위 집단에서는 특정영역의 지식/사고기능/기술의 이해 및 숙달 여부( $t = -5.178, p = .000$ ), 확산적 사고( $t = -4.958, p = .000$ ), 비판적/논리적 사고( $t = -2.173, p = .043$ ), 동기적 요소( $t = -3.157, p = .005$ ) 모두 통계적으로 유의한 향상을 보였다. 반면, 상위 집단에서는 특정영역의 지식/사고기능/기술의 이해 및 숙달 여부( $t = -3.346, p = .004$ ), 확산적 사고( $t = -2.821, p = .012$ )의 평균점수가 통계적으로 유의하게 향상되었으나, 비판적/논리적 사고( $t = -.203, p = .841$ ), 동기적 요소( $t = -.640, p = .531$ )는 평균점수가 향상되었으나, 그 차이가 통계적으로 유의하지 않았다.



**Table 7. The effect of education by group (creative problem-solving ability)**

	Sub-factor	M(SD)		t	p
		pre	post		
High initial score (n=18)	Creative problem-solving ability	3.839 (.482)	4.097 (.712)	-1.687	.110
	Understanding and attainment of skill/ability in specific areas	3.422 (.565)	3.844 (.701)	-3.346*	.004
	Divergent thinking	3.556 (.781)	3.989 (.698)	-2.821*	.012
	Critical/logical thinking	4.267 (.527)	4.311 (.900)	-.203	.841
	Motivational factors	4.111 (.558)	4.244 (.896)	-.640	.531
Low initial score (n=20)	Creative problem-solving ability	3.058 (.193)	3.470 (.446)	-5.269*	.000
	Understanding and attainment of skill/ability in specific areas	2.540 (.406)	2.990 (.447)	-5.178*	.000
	Divergent thinking	2.730 (.552)	3.290 (.575)	-4.958*	.000
	Critical/logical thinking	3.510 (.358)	3.760 (.480)	-2.173*	.043
	Motivational factors	3.450 (.481)	3.840 (.600)	-3.157*	.005

\*p < .05

### 4.3 계열별 수업 효과

#### 4.3.1 컴퓨팅사고력

사전, 사후 컴퓨팅사고력 점수의 차이에 관해 보다 심층적으로 분석하기 위하여, 전체 학생을 계열별로 구분하여 대응 표본 t검정을 실시하였다. 그 결과, 인문사회계열( $t = -3.534, p = .002$ )과 자연과학계열( $t = -5.640, p = .000$ ) 모두 학생들의 컴퓨팅사고력 평균 점수가 향상되었으며, 사전-사후 점수의 차이는 통계적으로 유의하였다.

**Table 8. The effect of education by major (computational thinking)**

	Sub-factor	M(SD)		t	p
		pre	post		
College of humanities and social science (n=19)	Computational Thinking	3.233 (.667)	4.011 (.869)	-3.534*	.002
	Computational concepts	3.175 (.658)	4.015 (.887)	-3.874*	.001
	Computational practices	3.224 (.692)	3.961 (.887)	-3.455*	.003
	Computational perspectives	3.386 (.780)	4.070 (.879)	-2.542*	.020
College of natural science (n=19)	Computational Thinking	2.940 (.904)	4.177 (.542)	-5.640*	.000
	Computational concepts	2.886 (.886)	4.256 (.594)	-6.063*	.000
	Computational practices	2.974 (.982)	4.145 (.555)	-4.697*	.000
	Computational perspectives	2.982 (.912)	4.035 (.543)	-5.319*	.000

\*p < .05

하위요인별로 대응 표본 t검정을 실시한 결과, 인문사회계열, 자연과학계열에서 3개 요인 모두 통계적으로 유의한 향상을 보였다. 구체적으로, 인문사회계열에서는 컴퓨팅사고력의 하위요인 중 개념( $t = -3.874, p = .001$ ), 수행( $t = -3.455, p = .003$ ), 관점( $t = -2.542, p = .020$ )의 평균점수가 향상되었고, 그 차이는 통계적으로 유의하였으며, 자연과학계열에서는 개념( $t = -6.063, p = .000$ ), 수행( $t = -4.697, p = .000$ ), 관점( $t = -5.319, p = .000$ )의 평균점수가 모두 향상되었고, 그 차이가 통계적으로 유의하였다.

#### 4.3.2 창의적 문제해결력

사전, 사후 창의적 문제해결력 점수의 차이에 관해 보다 심층적으로 분석하기 위하여, 전체 학생을 계열별로 구분하여 대응 표본 t검정을 실시하였다. 그 결과, 자연과학계열( $t = -4.545, p = .000$ ) 학생들의 창의적 문제해결력 점수가 향상되었으며, 사전-사후 점수의 차이는 통계적으로 유의하였다. 반면, 인문사회계열( $t = -1.731, p = .101$ ) 학생들의 창의적 문제해결력 평균 점수가 향상되었으나, 사전-사후 점수의 차이는 통계적으로 유의하지 않았다.

**Table 9. The effect of education by major (creative problem-solving ability)**

	Sub-factor	M(SD)		t	p
		pre	post		
College of humanities and social science (n=19)	Creative problem-solving ability	3.532 (.549)	3.758 (.719)	-1.731	.101
	Understanding and attainment of skill/ability in specific areas	3.011 (.544)	3.358 (.602)	-3.154*	.005
	Divergent thinking	3.295 (.746)	3.674 (.740)	-2.902*	.010
	Critical/logical thinking	3.958 (.691)	4.021 (.933)	-.301	.767
	Motivational factors	3.863 (.724)	3.979 (.931)	-.630	.536
College of natural science (n=19)	Creative problem-solving ability	3.324 (.506)	3.776 (.615)	-4.545*	.000
	Understanding and attainment of skill/ability in specific areas	2.905 (.764)	3.432 (.833)	-5.361*	.000
	Divergent thinking	2.947 (.797)	3.568 (.716)	-4.756*	.000
	Critical/logical thinking	3.779 (.452)	4.021 (.545)	-2.068	.053
	Motivational factors	3.663 (.472)	4.084 (.594)	-2.888*	.010

\*p < .05

하위요인별로 대응 표본 t검정을 실시한 결과, 자연과학계열에서는 3개 하위요인에서 통계적으로 유의한 향상

을 보였으나, 인문사회계열에서는 2개 하위요인에서 유의한 향상을 보였다. 구체적으로, 자연과학계열에서는 특정영역의 지식/사고기능/기술의 이해 및 숙달 여부( $t = -5.361, p = .000$ ), 확산적 사고( $t = -4.756, p = .000$ ), 동기적 요소( $t = -2.888, p = .010$ )의 평균 점수가 향상되었으며, 그 차이가 통계적으로 유의하였으며, 비판적/논리적 사고( $t = -2.068, p = .053$ )는 평균점수가 향상되었으나, 그 차이가 통계적으로 유의하지 않았다.

인문사회계열에서는 창의적 문제해결력의 하위요인 중 특정영역의 지식/사고기능/기술의 이해 및 숙달 여부( $t = -3.154, p = .005$ ), 확산적 사고( $t = -2.902, p = .010$ )의 평균점수가 향상되었으며, 그 차이는 통계적으로 유의하였으며, 비판적/논리적 사고( $t = -.301, p = .767$ ), 동기적 요소( $t = -.630, p = .536$ )는 평균점수가 향상되었으나, 그 차이가 통계적으로 유의하지 않았다.

## 5. 결론

본 연구는 학생들의 컴퓨팅사고력, 창의적 문제해결력을 신장시키는 데 중점을 두어 수업을 설계하고 그 효과를 검증하였으며, 학생들을 사전 검사 결과 및 계열에 따라 구분하여 수업의 효과를 분석하였다. 본 연구의 결과를 요약하고 정리하면 다음과 같다.

첫째, 컴퓨팅사고력, 창의적 문제해결력 신장을 위한 SW 수업은 학생들의 컴퓨팅사고력을 향상시켰으며 이는 기존의 선행연구[10][20][21]와 맥락을 같이 한다.

구체적으로, 컴퓨팅사고력 하위요인 중 개념, 수행, 관점 모두 평균 점수가 통계적으로 유의하게 향상되었다. 본 수업에서는 일상 생활에서 자신이 해결하고자 하는 문제와 관련된 융합프로젝트 계획서를 작성한 후, SW를 통해 이를 해결하는 융합프로젝트를 수행하는 것에 중점을 두었는데, 이러한 수업의 특징이 학생들의 컴퓨팅사고력 향상에 도움이 되었다고 할 수 있다.

더불어, 본 수업은 CT-CPS 모형을 기반으로 진행되었는데, CT-CPS 모형의 문제 인식 및 분석 단계에서 학생들의 흥미를 유발하기 위한 다양한 활동을 진행하였다. 구체적으로, 일상생활에서 컴퓨팅사고력을 활용하여 문제를 해결한 예시를 찾아보거나, 본인의 전공 분야에서 일어날 변화에 대해 생각해 보고 자신의 의견을 작성하는 활동을 진행하였는데, 이러한 활동이 학생들의 컴퓨팅사고력 신장에 긍정적인 영향을 주었다고 할 수 있다.

둘째, 컴퓨팅사고력, 창의적 문제해결력 신장을 위한

SW 수업은 학생들의 창의적 문제해결력을 향상시켰으며, 이는 기존의 선행연구[14][23,24]와 맥락을 같이 한다.

구체적으로 창의적 문제해결력의 하위요인 중 특정 영역의 지식/사고기능/기술의 이해 및 숙달 여부, 확산적 사고, 동기적 요소의 평균이 통계적으로 유의하게 향상되었으며, 이는 본 수업의 활동 내용에서 기인한 것으로 분석된다. 본 수업은 CT-CPS 모형을 기반으로 진행되었으며, SW를 통해 문제를 해결할 수 있는 아이디어를 다양하게 생각하도록 독려하였다. 구체적으로, CT-CPS 모형의 아이디어 구상 단계에서, 아이디어를 도출하기 위해 마인드맵, 브레인스토밍, SCAMPER 등을 선택하여 수행하도록 하였다. 이러한 확산적 사고 촉진 전략들은 특정 교과 내용과 상관없이 확산적 사고를 촉진하는 데 도움을 줄 수 있으므로[22], 학습자들의 창의적 문제해결력 향상에 영향을 미친 것으로 분석된다.

다만, 본 수업에서는 창의적 문제해결력의 하위요인 중, 비판적/논리적 사고 요인의 점수가 향상되었으나, 그 차이가 통계적으로 유의하지 않았다. 본 수업에서는 학생들의 비판적/논리적 사고 향상을 위한 토론의 과정이 포함되어 있었으나, 비중이 크지 않았다. 따라서, 기초 SW 수업에서의 토론을 확대하고, 토론 참여 독려 방안을 고려하는 것이 중요함을 시사한다.

셋째, 상위 집단과 하위 집단에 각각 대응 표본  $t$  검정을 실시한 결과, 하위 집단은 컴퓨팅사고력, 창의적 문제해결력이 모두 향상되었으나, 상위 집단은 컴퓨팅사고력에 서만 향상을 보였다. 하위요인별로도 차이를 보였는데, 하위 집단은 컴퓨팅사고력 및 창의적 문제해결력의 하위요인이 모두 향상되었으나, 상위집단은 컴퓨팅사고력과 창의적 문제해결력 하위요인 중 일부에서만 향상을 보였다. 이러한 연구 결과는, SW 교육을 통해 중, 하 집단 학생들의 컴퓨팅사고력이 신장되었으나, 상 집단의 학생들의 컴퓨팅사고력 향상은 통계적으로 유의하지 않았음을 보고한 선행연구[26]와 맥락을 같이 한다. 본 수업에서는, 프로그램을 구현하기 전, 아이디어를 추상화하고, 순서대로 표현하는 과정을 거치도록 하였다. 각 과정에 충분한 시간을 제공하였으며, 보완해야 할 부분이 있을 경우, 이전 단계로 돌아가서 대체, 결합할 만한 아이디어에 관해 다시 생각(SCAMPER)해 볼 수 있는 기회를 제공하였다. 즉, 프로그램을 구현하기 전에 문제해결을 위한 충분한 기회와 시간을 제공하였는데, 이러한 수업 방식이 문제해결에 상대적으로 시간이 더 필요한 하위 집단 학생들에

게 더 효과적이었던 것으로 분석된다.

또한, 본 수업에서는 실습 과정에서 문제 해결을 위한 도움을 점차 줄여가며 학생들이 스스로 예제를 해결할 수 있도록 하였다. 이러한 수업 방식은 모든 학생들에게 동일하게 적용되었으나, 문제 해결에 상대적으로 시간과 도움이 더 필요한 하위 집단 학생들에게 더 효과적이었던 것으로 분석된다.

넷째, 인문사회계열과 자연과학계열 집단에 각각 대응 표본  $t$ 검정을 실시한 결과, 자연과학계열은 컴퓨팅사고력, 창의적 문제해결력이 모두 향상되나, 인문사회계열은 컴퓨팅사고력에서만 향상을 보였으며, 창의적 문제해결력 점수가 향상되었으나, 그 차이가 통계적으로 유의하지 않았다.

구체적으로, 자연과학계열은 창의적 문제해결력의 하위요인 중 동기적 요소에서 통계적으로 유의한 향상을 보였으나, 인문사회계열은 동기적 요소가 통계적으로 유의하게 향상되지 않았다. 인문사회계열 학생들 또한 동기적 요소 점수의 향상을 보였으나, 자연과학계열 학생들에 비해 점수가 큰 폭으로 상승하지는 못하여 통계적으로 유의한 차이가 나타나지 않은 것으로 분석된다. 대학 교양 SW 수업에서는 비자발적 수강으로 인한 동기 부족 문제가 꾸준히 제기되고 있으며[5], 동기부여 없는 SW 교육은 고등사고력 신장 및 융합교육으로의 발전 또한 기대할 수 없으므로[2], 학생들의 동기를 높일 수 있는 교수-학습 전략이 고려되어야 할 필요가 있다. 특히, 관련성(Relevance)은 동기를 자극하고 유지하는 데 영향을 미칠 수 있으므로[33], SW와 전공과의 관련성이 적은 인문사회계열 학생들에게 수업의 필요성을 인지하도록 독려하거나 해당 수업의 유용성을 강조하는 전략이 필요하다. 나아가, 학습자는 자신의 전공과 컴퓨팅사고력의 연계 여부에 많은 관심을 가지고 있으므로, 수업 설계시 계열별 학습자 분석이 선행되어야 할 필요가 있다.

SW 교육은, 컴퓨팅사고력을 기반으로 일상생활의 문제를 창의적으로 해결할 수 있는 능력을 기르는 데 중점을 두어야 한다는 것[4]에 어느 정도 동기가 이루어지고 있다. 하지만, 대학생을 위한 SW 교육 프로그램이 부족하며[16], 구체적인 자료는 많지 않은 실정이므로, SW 교육에 대한 다양한 방법이 모색되어야 한다[16]. 본 연구는, 컴퓨팅사고력과 창의적 문제해결력을 향상시키는 수업을 설계하고자, 학생들이 스스로 융합프로젝트를 완성하도록 하였으며, CT-CPS 모형을 추가적으로 고려하여 수업을 설계하였다. 본 연구는 어떤 방법으로 교육을 진행

해야 하는지에 대해 구체적인 자료나 설계안이 부족한 대학 SW 교육에 시사점을 줄 수 있다.

하지만, 본 연구에서는 SW 교육의 효과를 컴퓨팅사고력과 창의적 문제해결력 변인을 통해 측정하였으므로, 융합 능력, 태도 등 다양한 변인을 통해 SW 교육의 효과를 측정할 필요가 있다. 또한, 본 연구는 단일 집단에 수업을 실시하고 그 효과를 검증하였으므로, 통계집단을 고려하여 연구를 설계하는 것이 필요하다. 더불어 본 연구는, 서울의 A 대학교 학생 38명을 대상으로 연구를 진행하여 연구 결과를 일반화하기에 어려움이 있으므로, 다양한 연령, 보다 많은 인원을 대상으로 한 후속 연구를 제안한다.

## REFERENCES

- [1] I. K. Chon, H. M. Lee & H. J. Cha. (2021). *Computational Thinking and Problem Solving*. Koyang: Infinitybooks.
- [2] G. J. Park & Y. J. Choi. (2018). Exploratory study on the direction of software education for the non-major undergraduate students. *Journal of Education & Culture*, 24(4), 273-292. DOI : 10.24159/joec.2018.24.4.273
- [3] Y. S. Jeong, J. S. Yoo, J. S. Lim & Y. K. Son. (2015). *SW education*. Seoul: Cmass.
- [4] Y. J. Jun. (2017). *The development and application of a CT-CPS(computational thinking-based creative problem solving) instructional model for the software education of new curriculum*. Doctoral dissertation. Korea National University of Education, Cheongju.
- [5] J. Y. Seo & S. H. Shin. (2020). Case Study on Problem-based Programming Classes in Software Education for Non-Computer Science Majors. *Journal of The Korea Society of Computer and Information*, 25(4), 213-222. DOI : 10.9708/jksoci.2020.25.04.213
- [6] H. J. Choi. (2011). The Programming Education Framework for Programming Course in University. *The Journal of Korean Association of Computer Education*, 14(1), 69-79. DOI : 10.32431/kace.2011.14.1.007
- [7] S. Dehnadi & R. Bornat. (2006). *The camel has two humps (working title)*. Middlesex University, UK, 1-21.
- [8] S. Y. Pi. (2020). Learning Effects of Flipped Learning based on Learning Analytics in SW Coding Education. Learning Effects of Flipped Learning based on Learning Analytics in SW Coding Education. *Journal of Digital Convergence*,

- 18(11), 19-29. DOI : 10.14400/JDC.2020.18.11.019
- [9] J. I. Kwon. (2019). Research of Computational Thinking based on Analyzed in Each Major Learner. *The Journal of Society for e-Business Studies*, 24(4), 17-30. DOI : 10.7838/jsebs.2019.24.4.017
- [10] M. J. Kim & H. C. Kim. (2018). Effectiveness analysis based on computational thinking of a computing course for non-computer majors. *The Journal of Korean Association of Computer Education*, 21(1), 11-21. DOI : 10.32431/kace.2018.21.1.002
- [11] M. J. Oh & M. R. Kim. (2018). Analysis of Effects of Scratch Programing Education to Improve Computational Thinking. *Journal of Korean Association for Educational Information and Media*, 24(2), 255-275. DOI : 10.15833/KAFEIAM.24.2.255
- [12] J. Y. Park. (2021). Analysis of the Effectiveness of Computational Thinking Classes for Improving Convergent Thinking Skills. *The Journal of Korean Association of Computer Education*, 24(1), 71-80. DOI : 10.32431/kace.2021.24.1.008
- [13] J. Y. Seo. (2017). A case study on programming learning of non-SW majors for SW convergence education. *Journal of digital convergence*, 15(7), 123-132. DOI : 10.14400/JDC.2017.15.7.123
- [14] Y. J. Jeon & T. Y. Kim. (2017). 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*, 20(4), 47-57. DOI : 10.32431/kace.2017.20.4.005
- [15] Y. J. Jeon & T. Y. Kim (2017). Analysis of Cognitive and Affective Effects in Elementary School Students' Software Class based on CT-CPS Model. *Proceedings of the Korean Association Of Computer Education*, 21(1), 25-28.
- [16] H. Park & Y. Jeon. (2021). A Design of SW Liberal Arts Class to Cultivate Creative Problem-Solving Ability and Learning Motivation of Non-SW Majors :Based on the CT-CPS Instructional Model. *Proceedings of the Korean Association Of Computer Education*, 25(2), 7-10.
- [17] J. M. Wing. (2006). Computational Thinking. *Communications of the ACM*, 49(3), 33-35.
- [18] J. M. Wing. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society*, 366, 3717-3725.
- [19] J. Y. Seo, S. H. Shin & E. H. Ku. (2018). A Study on Non-Majors Students' Perception of the SW Liberal Education in University. *Journal of Digital Convergence*, 16(5), 21-31. DOI : 10.14400/JDC.2018.16.5.021
- [20] S. J. Park. (2021). Analysis of the effects of non-face-to-face SW:AI education for Pre-service teachers. *Proceedings of the Korean Association Of Information Education*, 12(2), 315-320.
- [21] J. H. Lee & S. H. Lee. (2021). Analysis of the effect of non-face-to-face online SW education program on the computational thinking ability of students from the underprivileged class. *Journal of Creative Information Culture*, 7(4), 207-215. DOI : 10.32823/jcic.7.4.202111.207
- [22] Y. S. Cho, J. S. Sung & H. J. Lee. (2008). *Creative Education*. Seoul: Ewha Womans University Press: Seoul.
- [23] K. T. Kim & C. H. Lee. (2021). The Effect of Physical Computing Education Using Micro:bit on Creative Problem Solving Ability of Elementary School Students. *Journal of Korean practical arts education*, 34(1), 85-111. DOI : 10.24062/kpae.2021.34.1.85
- [24] E. G. Seo & Y. H. Shin. (2020). Development and Effect Analysis of Flipped Learning-Based Coding Education Course. *Proceedings of the Korean Association Of Computer Education*, 12(2), 315-320.
- [25] J. E. Nah. (2017). Software Education Needs Analysis in Liberal Arts. *Korean Journal of General Education*, 37(3), 63-89.
- [26] J. Moreno-Leon, G. Robles & M. Roman-Gonzalez. (2015). Dr. Scratch: automatic analysis of scratch projects to assess and foster computational thinking. *Revista de Educación a Distancia*, 46, 1-23.
- [27] Y. R. Lee & T. Y. Kim. (2020). CT-CPS based Informatics and Science Creative Convergence Class Design Using Physical Computing. *Proceedings of the Korean Association Of Computer Education*, 24(1), 189-192.
- [28] S. C. Lee & T. Y. Kim. (2018). Design of CT-CPS Based Programming Lesson Using NetsBlox for Elementary School Students. *Proceedings of the Korean Association Of Computer Education*, 22(2), 3-6.
- [29] H. S. Choi. (2014). Development of a Holistic Measure of Learning Effects in Robotics Program: Connecting Sociocultural Context and Computational Thinking. *Journal of The Korean Association of information Education*, 18(4), 541-548. DOI : 10.14352/jkae.2014.18.4.541
- [30] H. S. Choi & K. B. Kim. (2015). The Effects of Scratch Programming on Preservice Teachers: Assessment Utilizing Computational Thinking and

Bloom's Taxonomy. *Journal of The Korean Association of information Education*, 19(2), 225-232. DOI : 10.14352/jkaie.2015.19.2.225

- [31] S. H. Cho. (2001). *Development of creative problem solving test*. Korean Educational Development Institute.
- [32] E. Y. Jung. (2008). *Effects of Squeak Etoys based informatics education on the effects of Squeak Etoys based informatics education on elementary school student's creative problem solving ability*. Master's thesis. Korea National University of Education, Cheongju.
- [33] J. M. Keller. (2010). *Motivational design for learning and performance*. Springer, Boston, MA.

노 지 예(Jiyae Noh)

[정회원]



- 2017년 8월 : 이화여자대학교 교육공학과(교육공학박사)
- 2022년 3월~현재 : 덕성여자대학교 차미리사교양대학 조교수

- 관심분야 : SW 교육, 교수설계
- E-Mail : jynoh@duksung.ac.kr