

Analysis of Learner Satisfaction by Contents in Basic Software Education of College of Humanities

Seung-Hun Shin*, Joo-Young Seo*

*Assistant Professor, Da-San Univertisy College, Ajou University, Suwon, Korea

*Assistant Professor, Da-San Univertisy College, Ajou University, Suwon, Korea

[Abstract]

Based on the recent consensus on the need for software education, software education has become mandatory in universities in Korea. However, the suitability of class contents and the relationship between class contents and learner satisfaction has not been fully discussed. Therefore, in this paper, we analyzed the suitability of the contents used in the basic software education for the humanities students from the learner's perspective. For this purpose, three types of curriculum, which are 'computer science', 'usage of tools, and humanities in the digital world' and 'computational thinking', were compared using the lecture assessment questionnaire. As a result, we found that the learners evaluated positively over the curriculum that focused on the use of tools, the humanities approach, or the thinking ability rather than the theoretical contents of computer science. We also found that the subjects related to computer science could not give high satisfaction due to their unfamiliarity, relatively low academic value was given to the tool-centered curriculum, and fatigue was expressed in discussion-based thinking ability education.

▶ **Key words:** Basic Software Education, Humanities, Student-centered, Computational Thinking

[요 약]

최근 소프트웨어 교육의 필요성에 대한 공감대 형성에 따라 대학에서도 소프트웨어 교육이 의무화되었으나 학습자 관점에서의 콘텐츠 적합성 및 만족도는 여전히 논의가 필요한 문제이다. 따라서 본 논문에서는 인문계열 학생 대상 SW 교과에서 활용하는 콘텐츠의 학습자 관점 적합성을 살펴보았다. 이를 위해 '컴퓨터 과학', '도구의 활용과 디지털 시대의 인문학' 및 '컴퓨팅 사고'를 중심 주제로 구성된 세 가지 유형의 교과과정을 강의 평가 설문을 활용해 비교하였다. 분석결과, 학습자들은 컴퓨터 과학과 같은 이론적인 접근보다는 도구의 활용이나 인문학 측면에서의 접근, 혹은 사고력을 중심으로 한 교과과정을 더 긍정적으로 평가하였다. 또한, 컴퓨터 과학 관련 주제에 대해서는 생소함과 거부감 등으로 인해 만족도가 높지 않았고, 도구 중심 교육은 학문적 가치를 낮게 평가하였으며, 토론 중심의 사고 역량 개발 교육에서는 피로감을 표현하였다.

▶ **주제어:** SW기초교육, 인문학, 학습자 중심, 컴퓨팅 사고

-
- First Author: Seung-Hun Shin, Corresponding Author: Joo-Young Seo
 - *Seung-Hun Shin (sihnhsh@ajou.ac.kr), Da-San University College, Ajou University
 - *Joo-Young Seo (jyseo@ajou.ac.kr), Da-San University College, Ajou University
 - Received: 2020. 03. 19, Revised: 2020. 06. 10, Accepted: 2020. 06. 10.

I. Introduction

소프트웨어(Software, SW) 융합 기반의 지능정보기술이 직업, 고용 등 사회, 경제, 산업 구조 재편을 촉발할 것으로 전망됨에 따라 급변하는 시대를 주도하는 인재가 갖추어야 할 핵심 역량으로 디지털 기반의 컴퓨팅 사고(Computational Thinking, CT) 역량과 문제 발견 및 해결 능력, 그리고 융합 및 창의력이 새롭게 강조되고 있다[1-2]. 이에 따라 우리나라를 포함한 다수의 국가에서는 디지털 문제해결력의 기반이 되는 CT를 모든 학생에게 교육하기 위한 노력을 수행하고 있으며[1-3], 우리나라에서도 2018년부터 SW 교육을 의무화하는 등 정부 차원에서 SW 교육을 주도하고 있다[4]. 대학에서의 SW 교육은 과학기술정보통신부 주관의 SW중심대학 사업을 통해서 이루어지고 있다. 이 사업의 주요 구성 요소 가운데 하나는 전교생을 대상으로 하는 SW기초교육이며 이는 '타 전공 지식과 SW 소양을 겸비한 융합인재 양성'을 목표로 한다[5]. 따라서 이 사업에 선정된 대학에서는 학생의 전공과 무관하게 모든 학생이 SW 교육을 이수하고 있으며, 학교에 따라 2~9학점을 SW 교육에 배정하여 운영하고 있다[6].

초중고교의 경우 SW 교육을 위한 체계적인 준비 과정을 거친 후 교육이 시작되었으나[7-9] 대학의 경우에는 SW중심대학 선정과 동시에 교육이 시작되어야 했기 때문에 교육 대상의 특성이나 교육에 적합한 콘텐츠 및 이를 활용했을 때의 효과성 검토 등 교육 수행을 위해 검토되어야 하는 사항들에 대한 충실한 고찰이 부족한 상태에서 개별 대학 혹은 교수자의 선택에 따라 적응적으로 수행되었다[10-11]. 이후 다년간 교육이 수행되면서 대학생 대상 SW 교육의 목적, 교과과정의 구성 등 대학 SW 교육이 가져야 하는 정체성에 관한 연구와 함께 SW 교육에 대한 전공별 만족도나 학습 효과성에 대한 분석 등 교육 수혜자를 대상으로 한 다수의 연구가 수행되었고, 이를 통해 그간의 SW 교육 성과 확인과 함께 문제점 및 이를 해결하기 위한 다양한 방안들이 제시되었다[4][9-13].

학습 동기는 학습과 관련된 노력 및 태도에 큰 영향을 미친다[14]. 이는 학생들의 전공과 무관하게 필수로 시행되고 있는 SW 기초교육 또한 예외가 아니다. 하지만 SW 비전공자를 대상으로 한 연구결과에 따르면 학생들은 학습 동기 부족을 SW 교육의 가장 큰 문제점으로 지적하고 있다[11]. 동기 부족의 원인으로는 비자발적 선택으로 이루어지는 SW 기초교육 자체에 대한 거부감때문[4][15] 비전공자들에게는 생소한 학습 콘텐츠가 유발하는 상대적으로 높은 체감 난이도가 지목되었다[16]. 그러함에도 불구하고

비전공자들의 SW 교육 필요성에 대한 인식은 점차 긍정적으로 변화하고 있으며[17] SW 교육을 통한 성장은 비전공자에서, 특히 인문사회계열 학생에게서 더 높은 수준으로 나타난다는 연구결과[10][16]는 수강생의 만족도를 높일 수 있는 교육 콘텐츠 선정을 통한 학습 동기 부여의 필요성을 잘 보여주는 사례라 할 수 있다. 한편 비전공자 대상 SW 교육 모형과 교육에 사용된 콘텐츠에 관련된 연구에서는 CT 교육의 필요성을 인식시키는 수업을 프로그래밍 교육보다 앞서 이루어지도록 하거나[18] CT의 개념을 먼저 이해한 후 프로그래밍 언어로 연계[19]하는 것이 바람직하며 여기서 나아가 프로그래밍이나 컴퓨터 활용 능력 배양을 위한 교육보다는 문제에 대한 충분한 인식, 분석 및 효율적 절차 구성을 위한 학습 과정이 필요하다[19]는 등의 제언이 이루어졌다. 이러한 사실은 교육 콘텐츠의 중요성과 CT 자체에 대한 교육이 프로그래밍 교육에 선행되어야 한다는 인식은 공유되고 있으나 이를 위한 학습 콘텐츠는 추가의 논의가 필요함을 의미한다.

따라서 본 논문에서는 SW 교육과 정 반대 지점에 있는 인문학[13]을 전공하는 학생을 대상으로 프로그래밍 교육에 선행하여 개설되는 SW 교과를 서로 다른 콘텐츠를 활용해 운영한 사례를 강의 평가 결과를 활용하여 분석하고 이를 바탕으로 선행 교과에 적합한 콘텐츠의 유형을 살펴본다.

II. Preliminaries

1. Computational Thinking Education

디지털 시대를 살아갈 인재들에게 요구되는 네 가지 핵심 역량은 효과적 추론과 체계적 사고에 기반한 판단을 통한 문제 해결 능력, 명확한 의사소통 능력, 다른 사람과의 협업 능력, 그리고 창의적 사고 역량 및 혁신을 이끄는 구현 역량이다[20]. 이에 따라 우리나라 정부 또한 미래 핵심 인력 양성을 위해 SW 인재 양성에 힘을 쏟고 있다[2]. 하지만 컴퓨터 전공자만 SW 인재로 양성되어야 하는 것이 아닌 모든 인재가 정보화 시대를 주도할 수 있도록 해야 하므로 SW 비전공자들을 위한 컴퓨터 교육, SW 교육이 절실히 필요하게 되었다. 따라서 다양한 학문적 배경을 가진 SW 비전공자를 대상으로 하는 기초, 교양 교육과정 제공의 필요성이 제시되었다[12]. 한편 비전공자들의 경우 제한된 교양과목 범위 내에서 필요한 SW 교육을 이수해야 하는 한계가 있는데 이와 같은 환경에 적합한 교육이 CT 교육이다[21].

CT 교육이란 전반적인 ICT(Information and Communication Technology) 기술을 이해하고 이를 응용하여 주변의 문제를 해결할 수 있는 능력을 함양하는 교육이다[12]. SW 교육은 비판적 사고, 의사소통, 협동 능력을 바탕으로 한 CT 역량 향상을 목표로 하므로 여기에 프로그래밍 교육이 매우 효과적이라는 연구결과들이 제시되고 있다[22]. 또한, 프로그래밍은 컴퓨터 혹은 SW를 전공하는 사람뿐만 아니라 디지털 시대를 살아가는 모든 사람이 배워야 하는 기본 지식으로 알려져 있다[23]. 이전에는 21세기에 가장 필요한 역량 중 하나로 ICT 리터러시 혹은 디지털 리터러시를 선정하였지만, 현재는 ICT를 능숙히 사용하는 것만으로는 디지털 정보의 연결과 공유로 변화하고 있는 세상의 문제들을 해결하는 데 부족하다고 보고 소프트웨어의 사용뿐만 아니라 제작까지 할 수 있는 확대된 의미인 컴퓨팅 리터러시의 중요성이 강조되고 있다[24]. 하지만 SW 비전공 학생을 대상으로 프로그래밍 교육을 수행할 때 주목해야 할 점은 단순 기술 습득이나 도구를 사용하기 위해 프로그래밍을 배우는 컴퓨터 교육이 아니라[25] ICT 전반의 기술을 이해하고 문제를 해결할 수 있는 능력을 키우는 교육체제로 만들어야 한다[26]는 사실이다. 이를 위해 선행 연구에서는 컴퓨터 과학의 기초적인 내용을 활용해 이론 교육을 하고 스크레치나 파이썬을 문제 해결에 응용하거나[12][27], 널리 활용되는 워드 프로세서 및 스프레드시트 응용 프로그램을 컴퓨팅 사고 교육을 위한 도구로 활용[9], 혹은, 프로그래밍만을 활용해 컴퓨팅 사고 교육을 수행하였다[4][18].

하지만 실제로 SW 교육을 이수해야 하는 학습자 입장에서는 SW 기초교육이 어렵다는 의견과 본인 전공과의 연계성 부분에 의구심을 갖는 경우가 많다. 또한, 대학 입학 후 본인의 관심 분야에 맞춰 학습하고자 하는 학습자의 요구와 다르게 대부분의 대학에서 SW 교육을 기초 교양필수로 지정하면서 학습자들의 불만이 커지고 있다[4]. 이러한 사실은 학생들이 다제간 연계 전공 수업을 신청하면서도 본인의 전공과 이질적인 수업 내용에 대해 많은 두려움을 느끼고 있다[13]는 연구결과를 통해서도 확인할 수 있다. 따라서 모든 학생을 대상으로 하는 프로그래밍 교육을 설계하기 위해서는 전공을 목적으로 하지 않는 프로그래밍 초보자가 겪는 심리적 상황과 행동에 대한 분석과 이해가 반드시 선행되어야 한다[27]. 이에 근거한 연구결과[18]를 살펴보면 자연공학계열보다는 인문·사회계열과 예체능계열 학생들이 더 많은 어려움을 느끼고 있으며, 이 학생들은 그 이유로 '생소하고', '어려우며', '관련 전공

이 아님' 등의 복합적인 응답을 하였다. 또한, 흥미 향상, 사고력 향상, 성취도 향상 등의 평가 항목에서도 자연공학계열은 인문·사회계열 및 예체능계열과 유의한 차이를 보였다. 따라서 모든 계열의 학생들에게 CT 역량 향상을 위한 프로그래밍 교육을 적용하기 위해서는 CT 교육에 대한 필요성 제시를 위한 교육이 프로그래밍 교육에 선행되어야 하며 이 과정에서 프로그래밍 교육의 필요성과 프로그래밍에 대한 이해와 아이디어 개발, 고급 프로그램 언어로서의 자연스러운 커리큘럼 연계 및 향후 산업현장 혹은 실생활에서 어떻게 활용될 것인지에 대한 안내의 필요성을 제시하였다. 하지만 이처럼 선행 교과목의 필요성과 목표가 제시되었다고는 하나 교과목의 실제 구성이나 세부적인 운영 방법에 관한 연구는 충분치 않은 상황이다. 따라서 본 논문에서는 이러한 연구의 하나로 프로그래밍 교과목에 선행하여 운영하는 교과목에서 학습 콘텐츠가 학습자 만족도에 미치는 영향을 강의 평가 결과를 이용해 비교하고 이를 바탕으로 학습자 관점에서의 콘텐츠 적합성을 살펴본다.

2. Basic SW Education for Humanities

A 대학교는 SW중심대학에 선정되며 전교생을 대상으로 하는 SW 기초교육을 시작하여 다년간에 걸쳐 기초교양 교육으로서의 SW 교육을 수행해왔다. SW 기초교육을 시작하며 대학생이 가져야 하는 SW 기초역량을 '컴퓨팅 사고', '데이터 분석' 및 '프로그래밍' 역량으로 정의하였으며, 각 핵심 역량에 대해서는 다음과 같이 정의하고 있다.

- 컴퓨팅 사고: 컴퓨터를 활용한 창의적 문제 해결 역량
- 데이터 분석: 데이터로부터 지식과 통찰력을 찾는 융합 역량
- 프로그래밍: 창의적으로 문제 해결 방법을 고안하고 이를 자동화, 실체화하기 위한 기술 역량

이와 같은 핵심 역량에 대한 정의를 바탕으로 계열별로 필요로 하는 역량을 선정하여 각자의 요구에 부합되는 교육을 이수하는 전공 맞춤형 교육체계를 운영하고 있으며 모든 계열에서 6학점 이상을 필수로 지정하고 있다.

본 논문의 분석 대상인 인문대학의 경우에는 '컴퓨팅 사고'와 '데이터 분석' 역량을 선택해 교육을 수행하고 있다. 이 가운데 '데이터 분석'은 표면적으로는 R을 이용한 프로그래밍 교과목이나 전통적인 코딩 혹은 통계 중심의 교육이 아닌 데이터 과학을 이용해 다양한 유형의 데이터에 내재한 의미 있는 지식을 찾아낸 사례와 여기 사용된 방법을 보임으로써 학습자가 자신의 도메인에서 생성된 데이터를

조망하는데 요구되는 통찰력을 배양하는 것을 목표로 한다. 본 논문에서는 프로그래밍을 다루는 '데이터 분석'에 선행하여 운영되는 '컴퓨팅 사고' 교과에 적합한 콘텐츠를 탐색한다. 한편 '컴퓨팅 사고' 역량을 위한 교과에서는 SW와 미래 기술, 컴퓨터, 프로그래밍을 포함해 창의적 사고 역량까지 다양한 주제를 가용 주제로 간주하되, 하나의 수업 내에서 모든 주제를 한꺼번에 다루기보다는 학습자 특성과 필요성을 고려해 주제를 선별하여 콘텐츠를 구성하는 형태로 운영한다.

하지만 선행 연구에서는 학습자 관점에서 학습자가 실제로 필요로 하거나, 공감하는 주제가 어떤 것인지 분석되지 못했다. 프로그래밍 교과에 선행해 운영하는 '컴퓨팅 사고' 역량 교과에서 학습자들이 후수 교과에 자연스럽게 연계할 수 있도록 교육해야 한다[18]는 필요성과 목표는 제시하고 있으나 교과과정에서 활용하는 도구나 콘텐츠 및 운영 방법과 같은 세부적인 사항에 대한 제언은 이루어지지 못했다. 따라서 어떤 방법으로 SW 혹은 CT의 필요성을 학습자에게 전달할 것인지 그리고 프로그래밍으로 자연스럽게 연계시키는 방법은 무엇인지에 대한 논의가 필요하다. 이와 더불어 CT를 다루는 교과에서 CT를 어떤 방법으로 다룰 것인지도 고려해야 한다. 즉, CT가 다루는 개념 이해를 우선할 것인지 아니면 개념에 대한 언급은 배제하고 문제를 활용한 역량 배양에 집중할 것인지도 확인할 필요가 있다. 본 논문에서는 개별 주제의 적용 방법에 관한 연구에 선행하여 학습자의 학습 효능감과 만족도에 영향을 주는 콘텐츠는 어떤 유형인지를 그간 운영했던 교과과정에서 다른 콘텐츠를 비교하여 살펴보고자 한다.

III. Case Study

SW 기초교육을 수강하는 인문대학 학생 161명을 대상으로 학습을 희망하는 SW 교육 주제를 조사한 결과는 그림 1과 같다. 설문지는 개방형 서술 문항으로 구성되었으며 복수 주제가 기술된 응답으로 인해 총 174개의 응답이 제출되었다.

설문 결과, 학생들은 기초 코딩, C 등 프로그래밍 학습을 가장 많이 언급하였으며(26.4%), 뒤를 이어 엑셀과 파워포인트 등 오피스웨어가 24.7%, 영상 및 이미지 편집 도구가 12.1%였다. 이외에 인문학 관련 SW와 취업에 도움이 되는 주제 혹은 컴퓨터 기초 등의 기타 의견이 있었으며(10.3%), 희망하는 주제가 없다고 응답한 학생도 26.4%였다. 하지만 본 논문의 대상 교과는 프로그래밍 교

과에 선행하여 CT 교육에 대한 필요성 제시를 위한 것이고, 도구 활용 방법에 대한 교육은 현시점에서 적용 사례를 찾아보기 힘들기 때문에[28], 프로그래밍이나 오피스웨어 및 미디어 편집 SW를 중심으로 교과과정을 구성하는 것은 타당치 않다. 따라서 교육 내용은 전문가 협의를 통해 컴퓨팅 사고 함양에 더 큰 비중을 두고 있는 보편적 교육으로서의 컴퓨터 교육 추세를 반영하되 컴퓨팅 리터러시를 다룰 수 있도록 하였다.

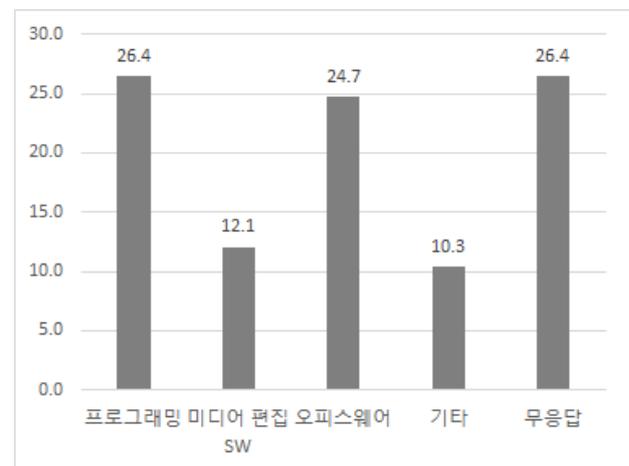


Fig. 1. Learners' Desired Contents

1. Curricula

A 대학교의 인문계열 학생을 대상으로 개설되는 컴퓨팅 사고 역량 배양을 위한 교과는 SW 계열, 인문계열 및 교양 교육 전문가의 연구결과를 바탕으로 현재까지 다음에 기술되는 세 가지 유형의 교육과정을 활용해 운영되었다.

1.1 Case CS - Introduction to Computer Science

SW 비전공자들이 수강하는 교양 교과과정에서는 컴퓨터 분야 전반을 이해시켜 통합적 감각을 익히게 하는 것이 필요하고[12] 컴퓨터 전공과목별 학습이 아닌 주요 개념을 통합하여 교과과정을 구성하는 전략이 필요하다[29]. 따라서 컴퓨터에 대한 기본적 원리를 익히고 컴퓨터와 관련된 다양한 주제를 탐색할 기회를 제공하여 컴퓨팅 사고력 함양을 목적으로 전문가 연구를 바탕으로 개발된 기존 '컴퓨터 과학' 과정[30]을 기초로 교육과정을 구성하였다. 다만, 운영할 SW 교과의 학습자는 기존 교육과정에서 고려하고 있는 학습자와 차이가 있으므로 기존 교육과정에서 제시하고 있는 주제를 활용하되 실제 교육 내용은 인문계열 학생들에게 적합하도록 난이도를 조정하고 일부 주제를 추가 혹은 변경하였다.

이 사례(CCS)에서는 표 1에 제시된 것과 같이 기존 컴퓨터 과학에서 다루는 주제에 비중을 두되 여기에 추가로

SW 및 컴퓨터와 관련된 시사성이 있는 주제와 사고력 배양을 위한 간단한 프로그래밍, 그리고 도구 활용 능력을 고려해 대표적 스프레드시트 도구 등을 다루었다. 또한, CT 역량 배양을 위한 팀 토론 및 발표를 포함하여 해당 교과에서 컴퓨터와 SW에 대한 이해와 창의적 사고 및 도구 활용 교육 등이 복합적으로 이루어질 수 있도록 하였다. 표2는 CCS 사례의 대주제별 비중을 보인다.

Table 1. Curriculum of the Case CS(CCS)

Week	Subject
1	The world of computers and SWs
2-3	Representation of data • Binary number, Data • Characters & civilizations • Logics, circuits
4	Computer components
5	Algorithm • Procedure of problem solving • Important algorithms
6	Block coding
7	Information system - Spreadsheets & DB
8-9	(Team) Block coding project
10	Abstract data types
11	Core of computer, Operating systems
12	Internet, connected world • Network technology • Firewall
13	World Wide Web
14	(Team) Internet-based application proposal
15	Future of computers & SWs

Table 2. Proportion of Topic in the Case CS(CCS)

Topic	Proportion(%)
Computer Science	53.3
SW Trend	20
Coding	20
Tool	6.7

1.2 Case HD - Humanity in Digital World

이 사례(CHD)는 컴퓨터 과학에 대한 이해보다 컴퓨터와 SW가 만들어 가는 세상과 그 안에서 살아가는 인간, 그리고 이를 활용할 수 있는 역량을 중심으로 교육 내용을 구성하였다. 해당 교육 내용은 A 대학교의 SW와 인문학 전문가의 논의 과정을 거쳐 정의되었다. 표 3에 제시된 내용을 살펴보면 컴퓨터 과학에서 다루는 이론 대신 컴퓨터를 도구로 활용하는 주체로서의 인간을 중심으로 교육 내용을 구성하여 학습자의 전공인 인문학에 대한 고려를 담았다. 또한, 컴퓨터와 SW가 학습자 본인 혹은 전공에 미

치는 영향과 관련성을 교수자에 의한 교육이 아닌 학습자 스스로 고찰할 기회를 제공하여 학습자 스스로 교육 내용이 가지는 의미를 찾아보도록 하였으며, 이외에도 인터넷이 보편화 된 현재를 살아가는 인간이 갖추어야 하는 윤리 의식에 관한 내용도 포함하고 있다. 이처럼 교과과정이 인간을 중심으로 SW를 조망하는 형태로 구성되었으나 이 교과가 SW를 다루는 기초 교과과정인 만큼 컴퓨터와 SW가 가지는 의미와 이를 도구로 활용하는 방법 및 CT 역량 배양을 위한 주제가 통합적으로 다루어지도록 하였다. 표 4는 CHD 사례에서 다른 주제별 비중을 나타낸다.

Table 3. Curriculum of the Case HD(CHD)

Week	Subject
1	How to use a computer
2	Advancement of computer
3	Computer and SW
4	(Team) Meaning of Computer and SW
5-7	Spread Sheets • Data Manipulation & create documents • Equations and functions • Representation of data
9-10	Computational thinking • Creative problem solving
11-13	Block coding
14	A new world of SW
15	Internet and human

Table 4. Proportion of Topic in the Case HD(CHD)

Topic	Proportion(%)
Humanities and SW World	28.6
Tool	28.6
Coding	21.4
CT	14.3
SW Trend	7.1

1.3 Case CT - Computational Thinking

이 사례(CCT)는 컴퓨터와 SW에 대한 이론과 도구 교육은 완전히 배제하고 순수하게 논리적·창의적 사고 훈련을 위한 내용으로만 구성하였다. 교육 내용은 A대학교의 SW와 교양 교육 전문가가 참여한 연구결과를 기반으로 구성되었다. CCT가 가지는 CCS 및 CHD와의 가장 큰 차이점은 교수자에 의한 직접적인 이론 위주의 교육은 최소한으로 줄이는 대신, 학습자가 스스로 팀 토론을 통해 CT에서 정의하고 있는 핵심 개념을 이해하고 이를 응용할 수 있는 역량을 배양하여 주어진 문제를 해결하도록 한 것이다.

표 5는 CCT의 교육과정을 보인다. CT 교육은 역량 배양을 위한 교육이므로 학습자는 학습하는 내용의 명칭을

굳이 인지할 필요가 없는 것으로 보았다. 따라서 CCT에서는 CT에서 정의하는 역량의 직접적인 언급은 배제되었으며, 강의 운영 방식을 안내하는 시간을 제외한 다른 모든 주치의 주제는 이전부터 학습자들이 인지하고 있는 쉬운 용어를 이용해 정의하였다.

학습되는 주제는 크게 학습자 개인이 CT의 주요 개념을 학습하는 부분과 학습된 내용을 바탕으로 팀 토론을 통해 문제를 해결하는 부분으로 나뉘어 있다. CT에 정의된 주요 개념을 학습하는 강의 초반부에는 '분해', '패턴인식', '단순화' 및 '알고리즘'을 예제를 통해 자연스럽게 학습하고, 제시된 문제를 개인적으로 해결하여 개념에 대한 이해를 확장하도록 했다. 이후 팀 토론 중심으로 운영되는 기간에는 중요한 문제 해결 수단인 분할정복, 탐욕 및 근사 알고리즘을 예제 문제를 활용해 학습하고, 이를 바탕으로 난도가 높은 문제를 팀원 간의 브레인스토밍을 이용해 해결한다. 또한, 모든 개별 주제에서 각 팀이 제안한 해결 방법을 다른 팀과 공유하도록 하여 하나의 문제를 해결하는데 다양한 접근 방법이 있을 수 있다는 사실을 이해할 수 있도록 하였다.

Table 5. Curriculum of the Case CT(CCT)

Week	Subject
1	Introduction to computational thinking
2	Division - Easier way to solve problems
3	Finding hidden patterns
4	Make it simpler, key-points only
5	Product of thought
6-7	(Team) Old man who moved a mountain
9-10	(Team) Do your best in everything
11-12	(Team) Too much or not enough
13-14	(Team) Finding problem worth solving
15	(Presentation) My way to solve the problem

2. Research Method

본 논문은 SW 기초교육을 이수하는 인문계열 학습자의 학습 콘텐츠에 대한 인식과 선호를 분석하기 위한 것이다. 연구 대상인 A대학교의 인문대학에서는 SW 기초교육을 위해 6학점을 필수로 지정하고 있다. 이 가운데 첫 번째 3학점 교과는 정원을 고려하여 통상 다수의 분반으로 구성되며 분반 선택은 학습자의 선택으로 이루어진다. 전체 분반은 동일한 교육 철학에 따라 수업이 운영되나, 교수자 재량으로 분반별로 세부 교육 내용이 다를 수 있으므로 본 논문에서는 세부 교육 내용이 동일한 분반만을 대상으로 분석을 수행한다.

2.1 Analysis Target and Procedure

연구 대상은 A대학교 인문대학 1학년에 재학 중인 학생으로 대학 입학 후 SW 기초교육을 처음 수강하는 학생이다. 분석 대상 전체 수강생 수는 270명이며, 교육과정별 수강생 수는 CCS 153명, CHD 81명, CCT 36명이다.

분석을 위해 사용된 강의 평가 설문은 A대학교에서 제작하여 개설된 모든 강의를 대상으로 활용 중인 것을 이용하며, 이 설문의 시행 시점은 종강 이후이다. 설문 결과 분석은 온라인 설문 조사 결과로 수집된 CSV (Comma-Separated Values) 형식의 데이터를 R 3.6을 활용해 수행하였다. 교과과정별 학습자의 인식과 만족도를 확인하기 위해 비모수통계(Kruskal-Wallis H)로 비교 분석하였으며, 또한 유의적인 차이를 보이는 사례 확인을 위해 사후 검정(Bonferroni)하였다. 또한, 개방형 질문에 대한 응답을 텍스트 분석을 통해 해석하였다. 텍스트 분석은 R 3.6에서 KoNLP와 tm 패키지를 이용하여 사용 빈도가 높은 단어와 응답 내 동시 출현 빈도가 높은 단어를 추출하는 형태로 수행하였다.

2.2 Questionnaire for the Lecture Assessment

A대학교의 강의 평가 설문지는 학습자 개인 의견을 서술하는 1개의 개방형 문항을 제외한 모든 문항이 5점 리커트 척도를 사용하는 형태로 구성되어 있으며, 문항은 '전혀 아니다'를 1로, '매우 그렇다'를 5로 코딩되어 있으므로, 평균이 5에 가까울수록 긍정적으로 인식을 하는 것으로 해석한다. 강의 평가 결과 분석은 설문지 문항 가운데 학습자의 체감 수업 난이도와 교과 구성 만족도를 파악할 수 있는 문항 6개를 선별하여 활용하며 교수자의 강의 태도 및 부수 문항은 분석에서 배제한다. 개방형 질문은 교육 내용 관련 의견만 수집하여 통계 분석의 결과를 해석하

Table 6. Questionnaire for the Lecture Assessment

	Question
R1	The lecture was conducted with academic interest and curiosity
R2	Class contents were organized and developed systematically.
R3	Teaching methods(lecture, discussion, demonstration, etc.) were appropriate for the class.
R4	The assignment appropriately supplemented the contents and helped to enhance the learning effect.
R5	Lecture materials were appropriate and helpful for learning.
R6	The difficulty and pace of the lecture were adequate.

는 데 활용한다. 개방형 질문의 해석은 R 프로그램을 활용해 높은 빈도를 보이는 단어를 선택하도록 하여 주관적 견해가 배제되도록 하였다. 분석에 사용된 개방형 질문에 대한 응답 수는 CCS 126개, CHD 65개, CCT 31개이다.

3. Analysis Result

분석대상인 세 가지 커리큘럼에 대한 학습자의 강의 평가 결과는 다음과 같다. 먼저 표 7은 CCS의 평가 결과를 보인다. CCS 학습자들은 '수업 내용이 체계적으로 구성되고 전개되었다'(R2)에 가장 긍정적인 응답(3.673)을 하였으며, 전체 학습자의 59%가 '그렇다'(37%) 혹은 '매우 그렇다'(22%)고 하였다. 한편 가장 부정적인 응답(2.908)은 '강의는 학문적인 흥미와 호기심을 갖도록 진행하였다'(R1)였으며 이 문항에는 전체의 34% 학습자가 '매우 아니다'(16%) 혹은 '아니다'(18%)로 응답하였고, 28%의 응답자만이 '매우 그렇다'(14%) 혹은 '그렇다'(14%)로 응답하였다. 또한, 학습자가 체감하는 강의의 난이도를 묻는 문항인 R6에 대한 응답도 부정적인 응답(36%)이 긍정적인 응답(33%)에 비해 높게 나타났다.

표 8은 CHD의 평가 결과를 보인다. CHD 학습자들은 '강의의 난이도 및 진행 속도는 적절하였다'(R6)에 가장 긍정적(4.198)인 응답을 하였으며, 전체 응답자의 75%가 '매우 그렇다'(38%) 혹은 '그렇다'(37%)를 선택하였다. 즉, 강의 난이도와 속도에 대한 학습자 의견은 CCS와 CHD에서 서로 반대되는 경향을 보인다. 한편, 이 설문에서 가장 부정적인 응답(3.938)을 보인 문항은 '강의는 학문적인 흥미와 호기심을 갖도록 진행되었다'(R1)였으며, 전체 응답자의 6%가 '아니다'로 응답하였다.

표 9는 CCT 평가 결과를 보인다. CCT에서는 '수업 내용이 체계적으로 구성되고 전개되었다'(R2)에 가장 긍정적(4.194)으로 응답하였으며, 전체의 83%가 '매우 그렇다'(36%) 혹은 '그렇다'(47%)고 응답하였다. 한편 가장 부정적인 응답을 보인 문항은 '강의의 난이도 및 진행 속도는 적절하였다'(R6)로 전체의 11%가 '매우 아니다'(3%) 혹은 '아니다'(8%)로 응답하였다.

응답 평균의 전체적인 경향을 살펴보면 수업 내용의 체계적 구성 및 전개(R2)는 모든 커리큘럼에서 다른 문항에 비해 상대적으로 높은 응답 평균을 보이고, CHD와 CCS에서는 학문적 흥미와 호기심(R1)이 가장 낮으며, 강의 난이도의 적절성(R6)의 경우, CHD는 가장 높은 반면, CCS와 CCT는 낮은 수치를 보였다. 즉, 학습자들은 도구 활용의 비중을 높이고, 도구 이외에도 다양한 주제를 다루는 CHD에 대해서는 쉽다고 인식하는 대신 학문적인 가치는

Table 7. Lecture Assessment Result of CCS

		1	2	3	4	5	Sum	Mean (Std. Dev.)
R1	freq	25	28	57	22	21	153	2.908(1.237)
	%	16	18	37	14	14	100	
R2	freq	6	8	49	57	33	153	3.673(0.999)
	%	4	5	32	37	22	100	
R3	freq	11	17	54	42	29	153	3.399(1.132)
	%	7	11	35	27	19	100	
R4	freq	17	19	55	37	25	153	3.222(1.193)
	%	11	12	36	24	16	100	
R5	freq	15	16	47	48	27	153	3.366(1.180)
	%	10	10	31	31	18	100	
R6	freq	24	30	49	27	23	153	2.967(1.269)
	%	16	20	32	18	15	100	

Table 8. Lecture Assessment Result of CHD

		1	2	3	4	5	Sum	Mean (Std. Dev.)
R1	freq	0	5	22	27	27	81	3.938(0.926)
	%	0	6	27	33	33	100	
R2	freq	0	3	14	33	31	81	4.136(0.833)
	%	0	4	17	41	38	100	
R3	freq	0	5	19	30	27	81	3.975(0.908)
	%	0	6	23	37	33	100	
R4	freq	0	4	12	37	28	81	4.099(0.831)
	%	0	5	15	46	35	100	
R5	freq	0	5	13	31	32	81	4.111(0.894)
	%	0	6	16	38	40	100	
R6	freq	0	5	15	30	31	81	4.198(0.954)
	%	0	6	19	37	38	100	

Table 9. Lecture Assessment Result of CCT

		1	2	3	4	5	Sum	Mean (Std. Dev.)
R1	freq	0	3	3	19	11	36	4.056(0.860)
	%	0	8	8	53	31	100	
R2	freq	0	0	6	17	13	36	4.194(0.710)
	%	0	0	17	47	36	100	
R3	freq	0	3	4	15	14	36	4.111(0.919)
	%	0	8	11	42	39	100	
R4	freq	0	2	8	15	11	36	3.972(0.878)
	%	0	6	22	42	31	100	
R5	freq	1	2	4	15	14	36	4.083(0.996)
	%	3	6	11	42	39	100	
R6	freq	1	3	8	13	11	36	3.833(1.055)
	%	3	8	22	36	31	100	

낮다고 평가하였으며, 컴퓨터 과학을 위주로 진행된 CCS의 경우 학문적 가치와 난이도 적정성 모두 낮게 평가하였다. 한편, CCT는 강의 난이도의 적정성(R6)은 낮은 응답

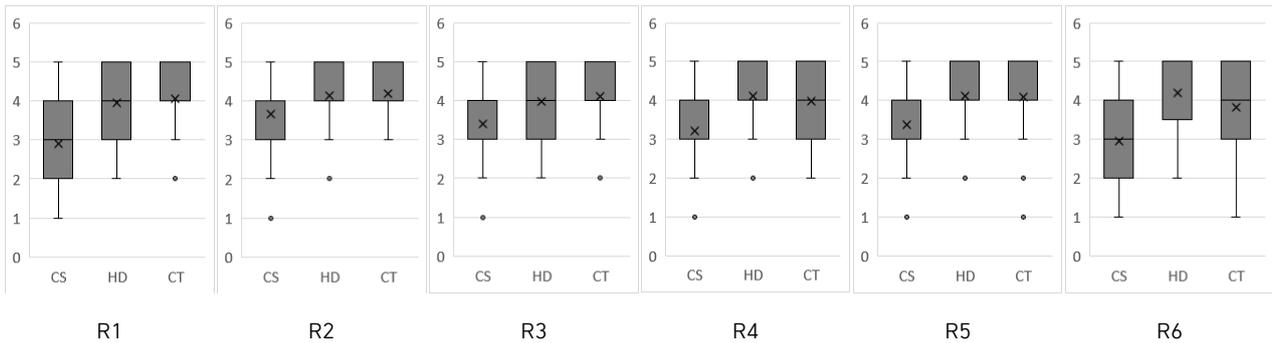


Fig. 2. Response Distribution by Question

Table 10. Analysis Result

	Kruskal-Wallis H		Bonferroni(p-value)		
	X ²	p-value	CS-HD	CS-CT	HD-CT
R1	50.7424	9.581×10 ⁻¹²	0	0	0.7998
R2	16.3252	2.851×10 ⁻⁴	0.0007	0.0067	1.0
R3	21.3208	2.346×10 ⁻⁵	0.0003	0.0005	0.6603
R4	35.9719	1.545×10 ⁻⁸	0	0.0008	0.7631
R5	28.3407	7.013×10 ⁻⁷	0	0.0006	1.0
R6	52.5037	3.972×10 ⁻¹²	0	0.0005	0.1997

평균을 보인 반면, 학문적 가치는 중간 수준의 수치를 보여 CCS와는 다른 특성이 있음을 알 수 있다. 한편, 교수 방법(R3), 과제(R4) 및 강의 교재(R5)에 대한 응답은 세 교과과정 모두에서 중간 수준의 응답 평균을 보였다.

분석결과, 표 10에 제시된 것과 같이 모든 문항에서 세 교과과정에 대한 평가 결과에 통계적으로 유의미한 차이 (p<.05)가 있는 것으로 확인되었다. 사후 검증에서는 모든 문항에서 CCS와 CHD 및 CCS와 CCT 사이에는 유의미한 차이가 있고(p<.025), CHD와 CCT 사이에는 유의한 차이가 확인되지 않아(p>.025) 학습자들은 CCS보다 CHD와 CCT에 대해 더 긍정적으로 인식하고 있음을 알 수 있다. 이러한 사실은 문항별 학습자 응답 분포를 표현한 그림 2를 통해 보다 명확히 확인할 수 있는데, CCS는 응답자의 50%가 2~4 구간을 선택한 반면, HD와 CCT는 3~5 구간을 선택했다. 한편, CCS는 학문적 호기심(R1)과 강의 난이도의 적정성(R6)에서, CHD는 학문적 호기심(R1)과 교수 방법의 적정성(R3)에서, CCT는 과제의 학습효과(R4)와 강의 난이도의 적정성(R6)에서 타 문항과 비교해 상대적으로 다양한 응답 분포를 보였다. 이는 CCS의 경우 CHD와 CCT에 비해 전체적인 강의 만족도가 낮기도 하지만 학습자 사이에서도 학문적 호기심과 강의 난이도 적정성에 대한 의견이 다양하게 분포하고 있음을 의미하며, CCT는 CCS보다 강의 만족도가 높으나 강의 난이도에 대한 다양한 의견이 있음을 확인할 수 있다. 이에 따라 CCS와 CCT의 강의 난이도에 대한 이해를 위해 개방형 질문의 응답을

텍스트 분석하여 학습자 의견을 살펴보았다.

표 11과 표 12는 각각, 개방형 질문에 대한 응답을 분석하여 추출한 '응답 내 동시 출현 단어'와 '사용 빈도가 높은 단어'를 보인다. 하나의 응답 내에서 동시해 출현한 단어를 살펴보면 CCS의 경우, '컴퓨터'는 '어려움', '인문학' 등의 단어와 함께 사용되었으며, '어려움'은 '재편' 혹은 '개편'과, '인문학'은 '대다수'와 함께 사용되는 빈도가 높았다. CHD에서는 '컴퓨터'를 중심으로 '교과과정', '적절', '문외한', '재미' 등의 단어가 함께 사용되었으며, 이와 별개로 '인문학'과 '스크래치'가 함께 사용되었다. 그리고 CCT에서는 CCS와 CHD보다 분산된 의견이 분산되어 있었는데, 다른 두 교과과정과는 차별되는 단어인 '알고리즘', '토론', '사고', '팀' 및 '비협조'와 같은 단어가 새로 등장하였고, '재미'와 '알고리즘', '어려움'이, '토론'과 '이해', '컴퓨터'와 '사고' 그리고 '팀'과 '비협조'가 함께 사용되었다. 또한, 모든 교과과정에서 학습자들은 '컴퓨터'라는 용어를 사용하고 있으나 부수적으로 사용되는 단어들은 교과과정에 따라 다양하게 변화된 것을 알 수 있다. 우선 CCS에서는 '인문대학' 혹은 '인문대학생'과 같이 학습자 소속이나 전공 분야를 나타내는 단어를 포함하여 '어려움', '필수', '기초지식'과 같은 단어가 높은 빈도로 사용되었다. 한편, CHD에서는 블록형 프로그래밍 언어인 'Scratch' 외에도 '학생', '재미' 등의 단어와 '인문학'이라는 단어 사용 빈도가 높았다. 그리고 CCT에서는 '강의'를 포함해 '알고리즘', '어려움', '토론' 등의 단어가 사용되었다.

이와 같은 분석결과를 다루어진 콘텐츠와 비교해보면 CCS에서는 SW가 인문계열에 맞지 않는 타전공이라서 생소하다는 인식이 강했으며, 이를 수강하기 위해서는 기본 지식이 필요하나 학습자 스스로는 그렇지 못하다는 평가를 하는 경향이 있어 전체적으로 교과과정을 어렵게 느끼는 것으로 확인되었다. 한편 CHD는 CCS와 CCT에 공통으로 나타난 '어려움'이라는 단어는 나타나지 않았다. 하지만, 자신이 '컴퓨터'에 대한 기초지식이 부족하다는 표

Table 11. Associated Words in the Response

CCS	CHD	CCT
<pre> graph TD computer --- difficult computer --- humanities difficult --- reformation humanities --- majority </pre>	<pre> graph TD curriculum --- computer computer --- suitable computer --- outsider computer --- interesting interesting --- humanities interesting --- scratch </pre>	algorithm — interesting — difficult discussion — understand computer — thinking team — uncooperative

Table 12. Frequently Used Words

Rank	CCS	CHD	CCT
1	computer	computer	lecture
2	humanities	scratch	algorithm
3	difficult	students	computer
4	mandatory	interesting	difficult
5	basic knowledge	humanities	discussion
etc.	unfamiliar, scratch, useful	final exam, outsider, complicated	data, useful, various

현이 등장한 것은 CCS와 동일하지만 CCS에서와는 다르게 이수한 교과과정이 재미있고 유용하다고 평가를 하였다. 한편, 컴퓨팅 사고 교육을 목적으로 활용한 블록 코딩의 경우, 인문대생에게 적절하지 않은 콘텐츠로 인식하는 경향이 있는 것으로 확인되었다. 마지막으로 CCT에서는 사고력 중심 교과과정의 정체성을 나타내는 '알고리즘'이라는 단어가 나타났다. 운영 방식을 표현하는 '토론', '팀' 등의 단어도 발견되었다. 한편, '어려움'은 CCS에서도 나타났던 단어이나 CCT에서는 CCS와는 다르게 전체 교과과정에서 요구한 '사고'에 대한 적응 관점에서의 '어려움'을 표현하고 있으며, 오히려 '재미'와 같은 긍정적인 단어와 함께 사용되었다.

IV. Conclusions

본 논문에서는 인문계열 학생 대상 SW 기초교육에서 프로그래밍 교과에 선행하여 운영되는 교과에서 활용하는 콘텐츠의 학습자 관점 적합성을 살펴보았다. 이를 위해 '컴퓨터 과학', '도구의 활용과 디지털 시대의 인문학' 및 '컴퓨팅 사고'를 중심 주제로 구성된 세 가지 유형의 교과과정을 강의 평가 설문을 활용해 비교하였다.

분석결과를 종합해보면 학습자들은 컴퓨터 과학과 같은 이론적인 접근보다는 도구의 활용이나 인문학 측면에서의 접근, 혹은 사고력을 중심으로 한 교과과정을 더 긍정적으로 평가하였다. 그 이유는 CHD의 경우, 도구 관련 콘텐츠 비율이 상대적으로 높아 학습자들로 하여금 교육의 실용성을 느낄 수 있도록 하고, CCT는 전통적인 컴퓨터 교육

형식에서 벗어난 사고력 개발을 위한 문제 풀이 과정을 어렵기는 하나 신선하게 인식하는 것으로 판단된다.

반면, CCS는 컴퓨터 과학에서 다루는 내용이 타 전공이라는 잘못된 인식에서 오는 생소함과 이를 필수로 수강해야 한다는 사실에서 기인한 거부감, 그리고, 기초교육임에도 불구하고 이를 학습하기 위해서는 별도의 기초역량이 필요할 것이라는 오해 등이 복합적으로 작용하고 있어 학습자에게 다가가기 쉽지 않은 주제인 것으로 판단된다. CHD에서 학습자들은 도구의 활용이나 SW와 인문학의 관련성과 같은 콘텐츠는 학문으로서의 가치가 낮다고 평가하였다. 그리고 CCT의 경우에는 상당 기간 지속되는 팀 단위 토론 수업이 주는 피로감과 타 교육과정에서 경험하지 못했던 사고력 개발 중심의 수업에서 어려움을 느끼며, 불필요하게 교육 내용과 컴퓨터의 연관성을 찾으려 시도하는 것으로 확인되었다. 한편, SW 기초교육을 위해 많이 사용되는 블록 코딩의 경우 인문계열 학습자는 그 필요성을 수용하기 어려워하였다.

이러한 결과를 바탕으로 처음으로 SW 기초교육을 수강하는 학습자를 대상으로 하는 교과과정에서 다루는 콘텐츠와 관련해 요구되는 사항을 정리하면 우선, SW 기초교육이 CT 역량 개발을 위한 교육 위주로 재편되고 있으나, 학습자들은 여전히 컴퓨터를 도구로 바라보고 있으므로 어떤 교육을 할 것인지 명확하게 목표를 설정한 후 이에 맞는 교육을 하는 것이 필요하다. 경우에 따라서는 도구를 활용한 사고력 교육이 필요할 수도 있다. 또한, 학습자들은 '실용성'과 '학문적 가치', 그리고, '어려움'을 명확히 식별하므로 학문적 가치와 교육의 효능감을 학습자들과 충분히 공감할 수 있는 주제의 식별이 필요하다 할 수 있다.

REFERENCES

[1] Presidential Advisory Council on Science & Technology, "Strategies for Competency Education on SW Convergence in the Fourth Industrial Revolution", pp. 1-2, 2017.
 [2] The Institute of Democracy, "Current Status and Improvement

- Software Education", 2018.
- [3] The Korean Academy of Science and Technology, K-12 Computer Science Education in Korea. KAST's 20th Anniversary International Symposium, 2014.
- [4] Jung-In Kwon, "Research of Computational Thinking based on Analyzed in Each Major Learner," The Journal of Society for e-Business Studies, Vol. 24, No. 4, pp. 17-30, Nov. 2019.
- [5] SW University Council, Introduction of SW University. SW University. <http://www.swuniv.kr/33>
- [6] SW University Council, Status of Selected SW Universities. SW University, <http://www.swuniv.kr/condition>
- [7] Ministry of Education, Ministry of Science, ICT and Future Planning, Plan for Human Resource Development for SW-centric Society, 2015.
- [8] Kyung-Sun Oh, and Seong-Jin Ahn, "A Study on Development of Educational Contents about Computational Thinking," Journal of Korean Association of Computer Education, Vol. 19, No. 2, pp. 11-20, 2016.
- [9] Kyung-Min Kim, "A Study on Information Literacy Education for Enhancing Computational Thinking," Journal of Korean Association of Computer Education, Vol. 20, No. 4, pp. 59-66, 2017.
- [10] Jung-Eun Na, "Software Education Needs Analysis in Liberal Arts," Korean Journal of General Education, Vol. 11, No.3, pp. 63-89, 2017.
- [11] Geum-Ju Park, and Young-Jun Choi, "Exploratory Study on the Direction of Software Education for the Non-major Undergraduate Students," Korean Journal of General Education, Vol. 24, No. 4, pp. 273-293, 2018.
- [12] Jeong-Eun Nah, "Analysis of Computational Thinking Learning Effect through Learner Observation," Korean Journal of General Education, Vol. 11, No. 5, pp.349-378, Oct. 2017.
- [13] Kyung-Sook Lee, "A Case Study for Constructing a Software Convergence Curriculum That Meets the Needs of Non-major Students," Culture and Convergence, Vol. 41, No. 5, pp.403-424, Oct. 2019.
- [14] Tae-Seob Shin, "An Analysis of Changes in Students' Achievement Goals in a Large-Scale General Education Class," Korean Journal of General Education, Vol. 8, No. 1, pp. 217-248, 2014.
- [15] Su-Jin Lee, "A Study on Designing a Class of Convergence Thinking based on Computational Thinking," The Korean Society of Science & Arts, Vol. 36, pp. 255-264. 2018. DOI:10.17548/ksaf.2018.12.30.255
- [16] Ju-Young Seo, "A Case Study on Programming Learning of Non-SW Majors for SW Convergence Education," Journal of Digital Convergence, Vol. 15, No. 7, pp. 123-132. 2017. DOI:10.14400/JDC.2017.15.7.123
- [17] Wan-Seop Kim, "A Study on the Recognition of Freshman on Computational Thinking as Essential Course," Culture and Convergence, Vol. 39, No. 6, pp. 141-170. 2017.
- [18] Mi-Ja Oh, Mi-Ryang Kim, "Analysis of Effects of Scratch Programming Education to Improve Computational Thiking," The Journal of Educational Inforamtion and Media, Vol 24, No. 2, pp. 255-275, 2018. DOI:10.15833/KAFEIAM.24.2.255
- [19] Eui-Sun Kang, "Structural Software Education Model for Non-majors - Focused on Python," Journal of Digital Contents Society, Vol. 20, No. 12, pp. 2423-2432, 2019.
- [20] National Education Association, "Preparing 21st century students for a global society: An educator's guide to the 'Four Cs'", 2012.
- [21] S. Bocconi, A. Chiocciariello, G. Dettori, A. Ferrari, K. Engelhardt, P. Kampylis, and Y. Punie, "Exploring the field of computational thinking as a 21st century skill," Proceedings of the International Conference on Education and New Learning Technologies, pp. 4725-4733, 2016.
- [22] In-Hwan Yoo, "A Study on SW Development Process for Increasing Computational Thinking." Korea Information Processing Society Transaction on Software and Data Engineering, Vol. 5, No. 2, pp. 51-58. 2016. DOI:10.3745/KTSDE.2016.5.2.51
- [23] Douglas Rushkoff, "*Program or be programmed: Ten Commands for a digital age*," OR Books, 2010.
- [24] Jung-Sook Sung, and Hyeon Cheol 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, 2015.
- [25] Shuchi Grover, "Learning to Code isn't Enough", EdSurge, an Independent Information Resource and Community for Everyone Involved in Education Technology. 2013.
- [26] Jeannette M. Wing, "Computational Thinking Benefits Society," Social Issues in Computing, <http://socialissues.cs.toronto.edu/index.html>, 2014.
- [27] Jung-Sook Sung, Soo-Hwan Kim, and Hyeon-Cheol Kim, "Analysis of Art and Humanity Major Learners' Features in Programming Class," The Journal of Korean Association of Computer Education, Vol. 18, No. 3, pp. 25-35, 2015.
- [28] Seong-Youn Hong, Joo-Young Seo, Eun-Hee Goo, Seung-Hun Shin, "Exploratory study on the model of the software educational effectiveness for non-major undergraduate students," Journal of The Korean Association of Information Education, Vol. 23, No. 5, pp. 427-440, 2019. DOI:10.14352/jkaie.2019.23.5.427
- [29] John F. Sanford, "Core concepts of computational thinking," International Journal of Teaching and Case Studies, Vol. 4, No. 1, pp. 1-12, 2013.
- [30] Korean Council for University Education, Computer Science Standard Curriculum, https://up.kcue.or.kr/eduClass_computer.jsp?le_nm=%C4%C4%C7%BB%C5%CD%B0%FA%C7%D0

Authors



Seung-Hun Shin received a B.S. degree in Information & Computer Engineering from Ajou University, Suwon, Korea, in 2000, and M.S. and Ph.D. degrees in Information & Communication Engineering from Ajou

University, Suwon, Korea, in 2002 and 2011, respectively. From September 2011 to February 2016, he was with the department of Software Convergence Technology, Ajou University as a Lecture Professor. Since March 2016, he has been with the Da-san University College, Ajou University as an assistant professor. His research interests include software testing algorithm, network intrusion detection, and mobile multimedia networking.



Joo-Young Seo received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Ewha Womans University, Korea, in 1993, 2001 and 2009, respectively. Dr. Seo joined the faculty of the Department

of Information and Computer Engineering at Ajou University, Suwon, Korea, in 2009. She is currently a Professor in the Dasan College University, Ajou University. She is interested in software education and software engineering with particular emphasis on software testing, embedded software testing and test automation.