

# A Study on Creative and Convergent SW Education Programs for improving Computational Thinking

Myung-Suk Lee\*

## Abstract

After the fourth industrial revolution came along, SW education to improve creativity and problem-solving ability began in elementary, middle, and high schools first and then in universities as well positively; however, research on the curriculum or what it has to pursue is not yet enough. Here, this study will investigate the current status of SW education provided in software-oriented schools operated in universities and also given as cultural studies in general universities and examines the curriculum or the standard plan for education. In most schools, it is operated as similar subject names, and diverse methods are tried on- and off-line to cultivate computing thinking skills. Also, to study SW education programs that can be operated in the general cultural courses of universities and find out how to utilize them, this author suggests the goal setting, educational contents, and teaching methods for SW education. As follow-up tasks, it will be needed to apply the suggested programs to the field and find out new evaluation methods in order to cultivate creative and convergent persons of ability.

▶ Keyword: Computational Thinking, SW Education, Creative Convergence, Hybrid Teaching Method

## I. Introduction

교육부는 개정교육과정을 통해 2018년부터 시행될 문·이과 통합형 교육과정 개편안에 SW교육을 포함시켰다. 변화하는 시대에 맞는 창의·융합형 인재를 양성한다는 것이 개정교육과정의 취지이다.

SW교육 개정 과정은 2019년부터 초등학교 실과과목에서 17시간이상 실시하고 2018년부터 중학교 정보과목에서 34시간이상 필수로 실시하고 2018년부터 고등학교 정보과목 심화선택과목으로 지정되어 있던 것을 일반선택과목으로 변경하여 실시한다. 또한 2020년까지 모든 초·중학교에 SW 관련 동아리가 1개 이상 운영된다고 한다.

이와 발맞추어 대학에서도 소프트웨어 중심대학을 지정하고 현재 14개 대학이 운영 중에 있다. 소프트웨어 중심대학은 반드시 운영해야할 사업요소 중, 대학교 전체 학생들에게 소프트웨어 기초 교과목을 1개 이상을 필수로 개설하는 것과 소프트

웨어 비전공자들이 소프트웨어를 복수/부전공 또는 연계전동을 하도록 장려하는 것을 두어 대학의 전체학생들에게 소프트웨어 기본을 교육시켜 주는 것이다. 이에 비전공자를 위한 SW교육을 각 학교마다 2-9시간을 유동적으로 운영하고 있었다. 또한 아직 소프트웨어 중심대학에 들어가지 못한 대학에서도 교양과목에서 1~2과목 SW관련 과목들이 개설되고 있으나 그 수는 아직 미비하였다. 지방 대학에서는 기존의 OA과목들을 그대로 운영되고 있는 학교가 많이 있었고 OA과목을 모두 없앤 학교 중에는 취업이라는 문턱에 걸려 다시 OA과목을 개설하는 학교도 있었다. 그러나 많은 대학에서 교양교육을 중심으로 그 필요성을 인지하고 있었다.

복잡한 사회의 다양한 문제를 해결하기 위해서는 컴퓨팅 사고력을 필요로 하고 컴퓨팅 사고력은 읽기, 쓰기, 셈하기와 같은 기본소양의 3R교육처럼 컴퓨팅 사고력이 여기에 추가되어

---

• First Author: Myung-Suk Lee, Corresponding Author: Myung-Suk Lee  
\*Myung-Suk Lee (mslee@kmu.ac.kr), Tabula Rasa College, Keimyung University  
• Received: 2017. 02. 23, Revised: 2017. 04. 25, Accepted: 2017. 06. 23.  
• This research was supported by the Keimyung University Research Grant of 2016

야 한다고 주장하고 있다[1]. 이와 더불어 대학교육의 학습방법에서도 주입식 강의가 아닌 토론식 강의, 학습자 주도의 수업으로 점차 바뀌고 있으며, 이러한 사회 흐름에 맞추어 인문학적 상상력과 과학기술 창조력을 갖춘 인재를 양성이 필요하다.

이에 본 연구에서는 SW교육 중심대학을 비롯하여 빠르게 확산되고 있는 SW 교육이 대학에서 어떻게 진행되고 있는지 실태를 조사하고 문제점을 파악한다. 또한 미래사회가 요구하고 있는 디지털 문제해결력을 기반으로 한 융합 사고력을 갖춘 인재를 양성하기 위하여 컴퓨팅 사고력 향상을 위한 창의·융합적 SW교육에 대한 프로그램을 연구하였다. 본 연구를 통해 컴퓨팅 사고를 증진시키고, 논리력, 창의력, 융합능력, 문제해결력 등을 키우는 것을 목표로 한다.

## II. Background

### 1. Related works

대학 내 학생들의 SW 교육에 대한 관심과 수요가 점차 증가하고 있고 대학에서도 그 수요를 반영하고 있다. 구체적인 예로 서울대 코딩교과목에 비전공자의 비율이 2012년에 9%였는데, 2015년에는 55%로 급증했다[22]. 그러므로 각 대학에서 비전공자도 컴퓨팅 사고력에 기반을 두어 해당분야에서 부가가치를 높이고 융합SW인재로 성장할 수 있도록 기회를 제공하는 노력이 필요하다.

현재 국민대는 국내 최초로 전체 학부생을 대상으로 SW교육을 의무화 하였고, 컴퓨터개론, 프로그래밍 언어 등 1년간 6학점 이상 SW과목을 수강하도록 하였다. 연세대에서는 컴퓨팅 사고력 과목을 2015학년도에 신설하여 시범적으로 운영하며 2016학년도 신입생 전체에게 필수교과목으로 실시할 예정이다. 또한 동국대학교에서도 2016학년도 신입생 중 인문·사회·예체능·자연계열 등 비전공자 전원을 대상으로 SW강좌 ‘소프트웨어와 미래사회’ 교과 수강을 의무화를 통한 SW융합형 교육을 목표로 하고 있다. 그 외에도 SW중점대학으로 선정된 모든 대학에서도 2016학년도 신입생 전체 대상으로 필수 교양과목으로 SW관련 교육을 실시하고 있다.

최근 SW교육과 관련된 여러 연구들을 살펴보면 다음과 같다. 임재열(2011)[2]은 프로그래밍 언어 교육현황과 개선을 제안하였다. 전공에서 이루어지고 있는 프로그래밍 교육 현황을 파악한 결과 사회나 산업체 요구에 미치지 못하며, 프로그래밍 언어 교육에 대한 문제점은 학생, 교재, 담당교수, 수강학생규모 그리고 수업시간 면에서 문제점을 지적하였다. 그 개선방안으로서는 실습효과를 고려한 선진대학 수준의 실습인원제한, 자기주도학습 지원을 위한 이러닝 시스템 구축, 학생중심의 창의적 교수법이 필요하다고 제안하고 있다.

류지성, 이성호, 김재원(2011)[3]은 IT 인재 양성을 위한 한국 대학교육의 과제를 살펴보면, IT 대학교육에서 교과과정의 표준화와 개발 학생에 대한 맞춤형 교과과정 설계가 모두 미흡

하며, 실습 인프라가 부족하여 양방향 소통보다는 일방향 강의로 학습효과가 저하되고 IT 교육의 질이 개선되기 어려운 실정이며 상당수 대학이 산업수요를 파악하지 못해 이에 기반한 교육이 이뤄지지 않고 있다는 연구 결과를 내놓고 있다.

차준섭(2015)[4]은 대학 SW 교육의 현장 적합도 분석에 기반한 현장 맞춤형 SW교육 과정 설계에 대한 연구를 하였다. 이 연구는 시장 상황에 따른 문제점을 해결하기 위한 대학의 SW인력양성 전략을 다음과 같이 9개로 제시하고 있다. 고급 소프트웨어 인력 양성, TA 및 산업체 전문 인력 활용, 팀 기반의 프로젝트 중심 교육, 교육 중심과 연구 중심으로 이분화, 현장 수요에 맞는 커리큘럼 구성, 교수 업적 시스템의 개선, SW과학 및 창업 관련 과목의 운영, 산학네트워크의 활성화, 재교육을 위한 전문 SW인력양성센터의 구축을 들고 있다. 이중 SW 과학 및 창업관련 과목의 운영을 살펴보면 대부분 대학의 SW교육이 공학 중심으로 이루어지고 있어 새로운 아이디어의 도출에 있어 한계를 드러내고 있다고 지적하고 다양한 영역의 학문과 융합할 수 있는 교육이 이루어져야한다고 주장하고 있다.

또한, 박성희(2016)[5]는 국·내외 대학에서 ‘컴퓨팅 사고력’ 교육에 대한 사례를 들고 CT-PBL의 모형을 Table 1과 같이 제안하고 있다. CT-PBL 모형은 대학교육에서 학습자를 위한 스캐폴딩 구조와 함께 프로젝트 실행 매 단계에서 어떤 실천을 할 수 있는지를 이론적 제시에 그치고 있다.

Table 1. ‘Basics for SW and Problem-Solving’ class

Step	Description and Activities
1	Asking questions and defining problems: Observation, relevant data search
2	Developing and using models: writing an inquiry report about detected problem with given form
3	Planning and carrying out investigations: Interview, web searching, survey, field investigation
4	Analyzing and interpreting data: Graph, table, chart, text
5	Using mathematics and computational thinking: Flow chart, use paper, presentation, block programming
6	Constructing explanations and designing solutions: Designing of the desired program using App inventor or Scratch within the program, writing a draft of block programming for the desired program
7	Engaging in argument from evidence: Debugging of the block programming of the program, completion of the program
8	Obtaining, evaluating, and communicating information: Presentation, demonstration of the completed program, feedback exchange about other's output(showcase)

이렇듯 기존의 연구들은 대부분 초·중·고 학생을 중심으로 연구가 진행되었거나, 이론적 연구 및 제안으로 연구가 진행되고 있는 실정이어서 구체적 교육방법이나 SW교육과 융합하는 커리큘럼을 제시하는 다양한 프로그램 개발이 필요하다.

### 2. Computational Thinking

컴퓨팅 사고력(CT: Computational Thinking)이란 용어는 2006년 Jennette Wing에 의해 처음으로 소개되었다. Wing은

컴퓨팅 사고력 정의를 ‘컴퓨터과학의 기초개념을 통대로 문제 해결, 시스템 설계, 인간행동의 이해를 내포 한다’고 하였으며 ‘아이들의 분석능력을 키우기 위해 CT를 읽고, 쓰고, 셈하기와 더불어 추가해야 한다’고 주장 하고 있다[6]. CT 용어는 주로 계산적 사고’, ‘정보적 사고’, ‘정보과학적 사고’, ‘컴퓨팅 사고’ 등으로 번역되고 있으나, SW교육과 관련된 논문이나 기사들을 보면 ‘컴퓨팅 사고력’ 또는 ‘컴퓨팅 사고’라는 말로 표현되어 있고, 교육가 발표한 소프트웨어 교육 청사진에도 ‘컴퓨팅 사고력’이란 표현을 하고 있다.

국내에서 컴퓨팅 사고력을 정의한 것을 보면 ‘인간의 사고와 컴퓨터의 능력을 통합한 사고’[7]로 요약하고 하고 있다. Peter Lee는 CT는 인간 지능을 확대하여 실제적인 적용을 할 수 있는 인간지능의 복잡도를 관리하거나 일을 자동적으로 처리하도록 하는 추상화 도구를 통해 확정하는 것이라고 정의하였다. CSTA(Computer Science Teachers Association)[8]와 구글의 Exploring Computational Thinking[9] 등에 따르면 다음과 같이 정의하고 있다.

CT는 문제를 풀이하는 과정으로 다음의 스킬들을 포함하여 컴퓨터 응용프로그램 개발에 핵심적인 개념이지만 수학, 과학, 인문학을 포함한 모든 분야의 문제를 푸는데 도움을 줄 수 있다. 문제를 해결하기 위하여 컴퓨터나 다른 도구를 사용할 수 있도록 문제를 형식화하기, 데이터를 논리적으로 조직화하고 분석하기, 모델이나 가상실험 같은 추상화를 통해 자료를 표현하기, 알고리즘적 사고(일련의 순서화된 단계)를 통해 해결책을 자동화하기, 가장 효율적이고 효과적인 단계들과 자원들의 조합을 찾는 것을 목표로 해서 가능한 해법들을 분석하고 구현하기, 이러한 문제해결 절차를 다양한 문제들로 옮겨보고 일반화하기, 결국 요약해보면 컴퓨팅 사고력이란 논리적으로 문제를 풀어가는 과정 또는 방법이라고 할 수 있다.

심리학에서 다루는 생활의 학습, 사고의 과정을 보면 이미 배워서 익숙해진 반응 양식에서 직관적으로 해결하기 어려운 부분에 직면했을 때, 우선 문제를 먼저 파악하고, 이 문제를 해결하기 위해 어떠한 행동이 유효한지, 어떻게 하면 목표에 이르는 수단과 방법을 찾을 것인가를 고민하게 된다. 그리고 여러 가지 시행착오를 거치는 동안 문제를 이해하는 통찰력이 생기고 문제를 해결할 수 있게 된다. 이러한 일련의 과정을 문제해결이다.

이러한 문제를 해결해 나가는 과정은 인류의 역사를 통해 보면 꾸준히 탐구되어 왔다. 과거에는 철학이 주로 이 역할을 담당했으나 컴퓨터 등장 이후 인간보다 훨씬 빠른 계산능력을 가진 컴퓨터라는 도구를 이용해서 문제 해결 방법을 찾기 시작했고 최근 인공지능이 등장하면서 문제는 더욱더 복잡해지고 복잡해진 문제를 해결하기 위해 컴퓨팅 사고력을 그 방안으로 내세우고 있다. 컴퓨터와 컴퓨팅 사고력의 등장 이후 물리, 화학 등 다른 분야에서도 이들의 도움을 받아 많은 변화가 일어났다. 이제 거의 모든 분야에서 소프트웨어를 사용하는 상황에서 컴퓨팅 사고력은 더욱 더 유용한 도구가 될 것이다.

박빈규 외(2014)[10]의 연구에 의하면 컴퓨팅 사고는 알고

리즘적 사고와 밀접한 관련이 있다는 연구가 나왔다. 알고리즘적 사고는 문제 상황을 해결하는 다양한 알고리즘 중 가장 효율적인 알고리즘을 찾고 이를 정교화하는 사고 과정으로 효과적이다. 학생들의 알고리즘적 사고력의 신장은 학생들이 일상 생활의 여러 문제들을 해결하는 데 있어 가장 기본적이고 핵심적인 내용과 절차를 찾아내는 능력의 신장으로 이어지기 때문에 변화하는 미래 사회를 살아갈 학생들이 필요한 폭넓은 사고력을 습득하게 할 수 있게 하는 반드시 익혀야하는 필수적 교양이라고 하였다.

### 3. Software Education

소프트웨어 교육(SW 교육)은 ‘코딩교육’이라고도 한다. 그러나 컴퓨터와 스마트기기 등을 단순히 활용하는 것에서 그치지 않고 알고리즘(문제를 해결하기 위해 구성한 일련의 순서화된 절차)과 프로그래밍을 통한 컴퓨터적인 사고력과 문제 해결능력을 키우기 위한 교육이다. SW 교육은 기존에 컴퓨터 활용 교육에서 문제점을 수정·보완하여 소프트웨어 기초를 소양교육을 통해 컴퓨팅 사고력을 신장하여 창의력과 문제해결력을 기르고자 하는 것을 목표로 하고 있다. 이를 위한 교과 내용으로 학생 중심 활동 학습과 SW Tool 활용 학습을 적용하고 적합한 프로그램 활동으로 제작 놀이 활동인 스크래치나 로봇활동, 엔트리 등 블록형 교육용 프로그래밍 언어를 활용하고 있다.

이러한 SW 교육이 필요한 이유는 컴퓨터적인 사고와 논리를 키우기 위해서이다. 앞으로 4차 산업혁명 시대에는 복잡하고 다양한 문제들이 발생한다. 이러한 문제를 해결하기 위해서는 기존의 암기식 교육방법으로는 복잡한 문제를 해결하기 어렵기 때문이다. 일반적으로 하드웨어를 움직이는 소프트웨어가 컴퓨터적 논리로 이루어져 있고, 이러한 프로그래밍 교육을 통해서 논리적 사고력을 키울 수 있다. 새 교육과정의 가장 큰 목표인 창의·융합형 인재 양성을 위한 방안으로 문제분석 능력, 논리적 사고력, 문제 해결 능력을 요구한다. 이러한 인재를 양성하기 위해서는 소프트웨어 교육이 필요한 것이다.

### 4. Prospective Research

‘SW 교육’ and ‘컴퓨팅 사고’ or ‘정보적 사고’를 키워드로 하는 연구를 검색하면 총 24건의 연구가 검색되며 이중 대학생을 대상으로 한 연구는 6건 정도의 연구 논문들이 검색되며 대부분 초·중·고 학생을 대상으로 한 연구이다. 그 중 전수진(2017)[11]은 CT 기반의 NDIS모형 설계연구에서 디자인중심 모델로 학습자의 요구를 분석하고 주어진 문제를 해결할 수 있는 디자인을 직접 설계하고 구현하며 이를 서로 공유하여 문제를 같이 해결해 나가는 교수학습모형으로 각 단계에 CT요소를 적용한 연구이다. 이 연구에서는 단순히 프로그래밍 기능을 익히거나 흥미 위주의 SW 교육을 넘어서 실생활에서 문제를 발견하고 CT를 기반으로 해결에 도전할 수 있는 기회를 제시해 줄 수 있는 교수학습방법에 대한 연구가 필요하다는 결론이다. 또한 류미영, 한선관(2015)[12]은 정보과학적 사고 기반 SW

교육 프로그램 개발 및 적용 연구에서 주제를 설정하고 주제에 따른 학습목표, 주제별학습내용을 개발하고, CT요소를 적용한 연구이다. 이러한 연구의 향후 필요한 연구를 보면 컴퓨팅 사고를 향상하기 위해서 체계적인 정보 교육과 프로그래밍 교육을 강화시킬 수 있는 방법을 모색하는 것이다. 이러한 연구들에서 보면 대부분 수업개발 단계인 계획→준비→개발→검토→적용 및 분석 단계에서 CT요소를 적용하는 연구였다. CT요소를 중심으로 어떻게 적용할 것인가에 대한 연구이며, 본 연구에서는 컴퓨터 사고력을 강화시킬 수 있는 방법과 CT를 기반으로 한 교수학습방법에 대한 연구라 할 수 있겠다.

### III. SW Education for Creative and Convergent

#### 1. Current Status of SW Education in University

세계에서 SW인재양성을 위한 경쟁이 심화되고 있는 가운데 국내 SW기업들은 여전히 원하는 수준의 역량을 갖춘 SW인재를 확보하는 데 어려움을 호소하고 있는 상황이다. 이에 국내에서도 미래창조과학부에서 소프트웨어 중심대학 14개 학교를 선정하여 운영하고 있다. SW 중점대학 확대는 올해는 6개를 더 선정하여 20개교로 확대할 예정이라고 한다. 총 14개 대학 중 서울·경기권이 10개 대학이고 지방대학이 4개 대학(경북대, 충남대, 부산대, 한국과학기술원)으로 구성되어 있다. SW교육의 목적은 개발자 양성이나 전공자만을 위한 것이 아닌 디지털 시대에 따른 컴퓨팅 사고력 함양과 융합 인재 양성에 있다. 또한 학생들이 SW와 다른 분야를 접목하여 새로운 가치를 창출하고, 미래 성장 동력이 될 수 있도록 하는 것이 SW교육의 궁극적 목표이다.

소프트웨어 중심대학은 4개의 사업요소가 있다. 첫째, 정원 100명 이상의 학생들 대상으로 소프트웨어 전공교육을 산업체 요구 중심으로 대폭 혁신하는 것, 둘째, 대학교 전체 학생들에

Table 2. System Environment

No.	University name	Year	Subject	necessary* Completion time	Etc *Parentheses indicate the (credits / class hours).
1	gachon Univ.	2015	SW basic, SW convergence education	2	Open source software-based operations Not-major 4 affiliation groups 6 course completion(software-centric world, programming start, computational thinking, hand-held programming, software ecosystem, application programming) Operate preliminary university Blended Running
2	kyungpook national Univ.	2015	SW and problem solving basic	3	Humanities(Software Thinking Techniques), Natural Sciences(social network), Arts & Physical(Introduction to Culture Technology), choice 1-2 courses out of 10 courses (3/3)
3	korea Univ.	2015	Computational thinking	4(3)	Step 1 : Operate online / offline SW education program before enrollment, Coding education and computational thinking courses essential through common education system Step 2 : SW coding education for non-majors, such as humanities and society, using core education system(scratch, python) (Software Activities, Computer Programming, Data Structure, Computer Programming Practice, Data Structure Practice, Introduction to Telecommunications, Computer Language and Practice, Computer Science and Practice
4	sogang Univ.	2015	Computational thinking	3	3 credits required for all undergraduate basic SW courses and up to 6 credits for all students SW courses With basic literacy training(3/3).
5	sungkyunkwan Univ.	2015	Computational thinking and SW coding, Problem solving and Algorithm	4	Completion of 1-2 courses per semester
6	sejong Univ.	2015	SW basic, Introduction to programming	3	Entertainment SW, Global Media SW 2-3 tracks completed (Arts & Physical major).
7	ajou Univ.	2015	SW basic education	6	
8	chungnam national Univ.	2015	Computational thinking, Programming basics	3	SW course experience events for non-major(Drone, VR), Must complete one course out of Python SW Education, Basic Coding SW Scratch Education, Mobile App Programming SW Education for
9	kookmin Univ.	2016	Computer programming1, 2	6	Flipped learning, Completely learning(3/3)
10	dongguk Univ.	2016	Software and future society	2	2/2
11	pusan national Univ.	2016	SW basic education, SW convergence education	3	Humanities/Society(Big data utilization technology), Natural Sciences/Business(Python programming, Statistics SW, Big data utilization technology, Pintech), Arts(Design SW utilization technology, Physical computing application) (3-6 hours completion)
12	seoul women's Univ	2016	SW and Creative thinking	3	Programming non-curriculum requirement(300 hours) Flipped learning type, dormitory type, Ada school lab operation
13	kaist	2016	Programming language, Elice SW practice	9	Elice SW practice education, programming language, Package Usage Attribute Training
14	hanyang Univ.	2016	Software	2	Open source SW education for non-professional SW education(2/2)

\* In time () is the same as class time and grade

계 소프트웨어 기초 교과목을 1개 필수로 개설하는 것, 셋째, 소프트웨어 비전공자들이 소프트웨어를 복수/부전공 또는 연계 전공을 하도록 장려하는 것, 넷째, 지역사회 (즉, 초·중·고 학생 및 일반인)대상으로 소프트웨어 기본을 교육시켜 주는 것이다.

Table 2는 소프트웨어 중심대학의 비전공자를 위한 SW교육 과목명과 필수 이수시간을 조사하였다. 모든 소프트웨어 중심 대학의 과목운영을 조사한 결과 4개 사업요소를 잘 반영하고 있었다. 그중 SW 교육을 비전공 전체학생을 대상으로 한 기초 교양 필수과목으로 2~9학점으로 운영되고 있었다. 특징은 기존의 컴퓨터 실습과목은 4시간 수업에 학점이 3학점(3/2/2) 또는 3시간 수업에 2학점(2/1/2)으로 운영되는 것이 일반적이었다. 그러나 SW교육 수업에서는 대부분 프로그래밍 실습수업이 운영되고 있었고 수업시수와 학점이 대부분 3/3 또는 2/2로 일치함을 보였다. 그만큼 새로 시도 되는 수업으로 커리큘럼 개발부터 많은 노력이 필요하기 때문이라고 하였다.

기타에서 특징들을 조사하였는데 1학년이 되기 전 입학자를 위한 예비대학에서 SW교육이 온/오프라인으로 진행되는 곳이 많았고, A학교는 예비학생이 SW교육을 온라인으로 이수한 결과가 있어야 학교 홈페이지 접근과 수강신청이 가능하도록 해 모두가 필수로 이수할 수 있도록 하고 있었다.

과목명은 모두 비슷비슷 하였고 SW 코딩, 문제해결, 컴퓨팅 사고력, 알고리즘, 프로그래밍, 융합 이라는 키워드로 압축이 되었고, 대부분 실습수업으로 이루어지고 있었으며 몇 개의 학교는 온라인을 이용한 이론 수업도 진행되고 있었다.

B대학에서 이루어지고 있는 'SW와 문제해결 기초' 교과목의 커리큘럼을 살펴보면 Table 3과 같다. 프로그램 실습을 위한 기초들로 되어 있다. 이론적 설명으로 되어 있는 부분도 있으나 실습을 통해서 학습자들이 창의적 사고할 수 있도록 구성되어 있었다.

Table 3. 'Basics for SW and Problem-Solving' class

time	Lessons and Instructional Objectives	Assignment and Research Issues
1	01. Course Introduction - Class guide - Computer overview - Flowchart, pseudo code and program overview Exercise 1 Obtaining official roots using dichotomy	Formulas and assignments
2	02. Programs and software Practice 02. Finding the result of a sequence	Assignment and iteration
3	03. Flowchart and pseudo code 1 Practice 03. Finding the result of combination	Condition and iteration
4	04. Flowchart and pseudo code 2 Practice 04. Making a combination of coins	Multiple conditions and multiple iterations
5	05. How to solve the given problem? Exercise 05. Find the sum of the widths of repeated triangles	How to think

6	06. How to approach the problem 1 Practice 06. Putting coins in and out of a structure with only one coin at the top	Stack
7	07. How to approach the problem 2 Practice 07. Putting and subtracting values from the front or the front of a long ladder-like structure	Queue
8	midterm test	
9	08. How to approach the problem 3 Practice 08. Replace rows and columns of elements in a matrix	Matrices and arrays
10	09. Experiential Problem Solving Methods Practice 09. Create a way to add and calculate like humans	A.I
11	10. Graph Exercise 10. Ensure that the sum of any row or column in a 4X4 matrix is the same	Array and Graphs
12	11. Connect all the points Practice 11. Finding the path from the center cell to a specific cell	Connect all the points
13	12. Shortest path search 12. Find the shortest route from a specific point on the map to a specified destination	Find the shortest path
14	13. Division and conquer Exercise 13. Find the name and number you entered in the telephone book listed by number Practice 14. Classify data Practice 15. Summarize the various data classified	Division and conquer
15	Final exam	

또한 전체 국립대를 대상으로 교육대학을 제외한 26개 전체 대학에서 SW교육에 대한 내용을 조사하였다. 그 결과 아직 SW중점대학이 아닌 경우에는 대부분 겉으로 드러나는 SW 중심 교육은 실시되지 않고 있었다. 다만 음성적으로 2개 학교는 교양 필수로 프로그래밍 실습 및 기초를 필수로 개설하고 있었고 학점/시수는 일반적으로 3/2/2/(학점/이론/실습)이나 2/1/2로 구성되어 있었다. 그 외 7개 학교 정도는 교양과목에서 한두 과목이 개설되어 운영되고 있었다. 아직 많은 학교에서 OA과정을 아직 그대로 운영하고 있는 것을 알 수 있었다. 실제 OA과정도 모두 없애기에는 학생들이 취업하는데 당장은 필요한 과목이라 모두 SW교육으로 옮겨가기에 위험하다고 볼 수 있다.

OA수업을 지속할 것인가 아니면 현 시점에서 필요 없는 수업이므로 수업의 방향전환을 할 것인가를 측정하기 위하여 기업을 대상으로 신입사원을 뽑을 때 필요한 역량을 설문조사하였다. 기업선정은 대기업이 아니라 학교와 산학협력을 맺고 있는 중소기업 이하 소규모 기업을 대상으로 하였다. 그 결과 규모가 적은 기업일수록 OA활용도가 높다고 답하였으며, OA중 엑셀, 파워포인트, 한글, 정보기술 전반적인 이해 능력 순으로 중요도가 나타났다. 일반적으로 사원이 갖추었으면 하는 자격증은 컴퓨터활용능력1급, MOS마스터, 워드프로세스 순으로 나타났다. 응답한 대부분은 기업이 컴퓨터 활용 능력에 대한 승진 인사규정이 없으며, 연수 프로그램도 없기 때문에 필요한 활용 능력은 사원 스스로 습득해야한다는 의견을 주었다. 기업에서는 OA과목 중 엑셀, 한글, 파워포인트 순으로 필요성을 강조했다. 현재는 4차 산업혁명시대를 준비해야하고 현재의 상황에서

취업준비를 갖추어야하는 과도기적 상황이기 때문에 어쩔 수 없이 두 가지를 양립하여 진행하는 것이 학생들에게 당장의 취업과 미래를 설계하고 준비하는데 도움이 될 것이다.

**2. Application of Hybrid Teaching Method**

새 시대가 요구하는 창의·인재를 양성하기 위해서는 교육과정도 재정비하여야 하며, 교수방법 또한 많이 연구되어야 한다. 최근에 연구된 하이브리드 교수법[13]은 기존의 플립 러닝이나 블렌디드 러닝의 장점들을 이끌어내고 단점들을 보완하여 제안된 학습법이다. 두 교수학습법의 장점은 학습자의 자기주도적 학습이 가능하다는 것이고, 단점은 학습 밀도의 부족, 교수 피드백의 부족, 수업 집중도 부족 등을 꼽을 수 있다. 이에 하이브리드 교수법을 온·오프라인 통합교육과 경험/성찰 단계를 두어 형식학습과 무형식학습을 통합하는 것으로 정의하고 있다. 이러한 하이브리드 교수법을 SW교육 수업에 활용하고자 한다. SW 교육 수업 프로그램에서 단계별로 학습자/교수자 역할은 Table 4에서 제시하였다.

Table 4. Step-by-step learner / instructor role

	Learner role	Teacher role
Knowledge acquisition step	<ul style="list-style-type: none"> <li>- acquire SW basic education</li> <li>- Acquire basic knowledge through lectures on preparatory lesson / review based on online</li> </ul>	<ul style="list-style-type: none"> <li>- Links MOOC videos related to the lecture goal or MOOC videos production</li> <li>- Curator, Teaching role</li> </ul>
Thinking expansion step	<ul style="list-style-type: none"> <li>- Based on the topics presented by the instructor, they actively participate in various active learning activities such as Flipping Learning, Action Learning, and Project / Problem Based Learning.</li> </ul>	<ul style="list-style-type: none"> <li>- Presents learner-participative problem for the whole curriculum</li> <li>- Instructional direction in PBL problem solving process</li> <li>- Guidance, role of facilitator</li> </ul>
Experience / reflection step	<ul style="list-style-type: none"> <li>- Problems in real life or Field</li> <li>- Presenting creative ideas and solving problems based on what you have learned</li> <li>- Improving creativity and problem-solving ability</li> </ul>	<ul style="list-style-type: none"> <li>- Present the problem finding topic so that the contents related to the class can be experienced in the field</li> <li>- Role to help solve the problem</li> <li>- Mentor role</li> </ul>

첫째, 지식습득 단계로서 기초학습이 이루어지는 단계이다. 이 단계에서 학습자는 온라인을 통하여 기초 지식을 습득한다. 교수자는 수업 목표와 관련된 MOOC 영상을 링크하거나 또는 영상을 제작하여 온라인상으로 제공하며 큐레이터나 티칭 역할을 담당하게 된다.

둘째, 사고확장 단계에서는 지식습득 단계에서 기초학습이 이루어지고 난 후 학습자는 FL, PBL 등 다양한 AL을 통해서 사고를 확장하는 단계를 거치게 된다. 교수자는 교수법이 잘 진행될 수 있도록 커리큘럼에 맞는 학습자 참여형 문제를 제시하고 문제해결과정도 잘 체크하여야하며, 특히 프로젝트 기반학

습은 학습자중심학습을 촉진하는 방법으로 효과적이다. 교수자는 정보를 전달하기보다 유연한 코치가 되고, 학생들과 브레인 스토밍을 하고, 탐구적인 행동을 모델링하는 가이드 역할해야 한다.

셋째, 학습자가 수업내용을 실생활에서 적용해보는 경험/성찰학습 단계이다. 경험학습은 학습자들이 콘텐츠의 의미, 관련된 아이디어, 이해를 높이기 위해 과거의 경험 또는 전공과 연결 짓고 아이디어를 창안하는데 초점을 맞춘다. 학습에 대한 지속적인 동기부여를 위해 학습자들은 교육과정과 실세계 사이의 명확한 연관성을 찾고, 실세계에서 경험하고 새로운 아이디어를 창조하는 역할을 한다. 이때 교수자는 멘토 역할을 한다. 학습자는 학습한 지식을 활용함으로써 창의력과 문제해결 능력을 키워 완전학습을 할 수 있도록 한다.

**3. SW Education Program**

먼저 SW교육 프로그램을 운영하기 위한 지침을 마련하고자 한다. 소프트웨어 교육 운영지침은 컴퓨팅 사고력을 가진 창의·융합인재의 양성을 목표로 하여 전공별 소프트웨어 교육의 연계성을 유지하고 단순한 정보통신기술 활용 교육을 넘어 타 전공과의 융합을 유도하고 소프트웨어로 새로운 사회의 가치를 창출하는 컴퓨팅 사고력을 갖춘 인재를 양성하는데 목적을 두어야 한다.

개발 방향은 우리의 일상생활의 모든 영역에서 소프트웨어가 활용되듯이 우리 생활 주변에서 발생하는 또는 교과목에서 발생하는 문제를 해결할 수 있는 역량을 기를 수 있어야 한다는 데 운영지침의 기본적인 방향을 담고자 하였다. 따라서 교육부에서 발표한 초·중·고 SW 교육 운영 지침[14]을 기반으로 하여 목표, 내용, 교수방법을 작성하였다.

**1.1 Set the Goal**

목표 영역을 Table 5와 같이 생활과 소프트웨어, 알고리즘과 프로그램, 컴퓨팅과 문제해결 영역으로 나누었다.

Table 5. Set the Goal

Domain	Goal
Life and software	Understanding the change of life by software and comprehension of technology fused with computing technology
Algorithms and Programming	Experiment and design with algorithm and programming
Computing and problem solving	Discover real life problems based on computing thinking and Solving convergence problems related to major

컴퓨팅 사고의 7단계를 보면 문제분석단계, 데이터 수집과 표현단계, 분해단계, 패턴인식단계, 추상화단계, 알고리즘단계, 평가단계로 되어 있다. 여기에서 자주 등장하는 단어가 알고리즘, 구조화, 추상화, 모델링, 시뮬레이션 이라는 단어가 자주 반복된다. 이것을 정리해보면 알고리즘이란 문제를 해결해나가는 과정을 알고리즘이라고 할 수 있다. 일반적으로 라면에 많이 비

유한다. 라면 끓이는 방법을 보면 다음과 같다. 라면을 준비 한다 - 냄비에 물을 붓는다(입력) - 냄비에 물이 끓는가?(조건문) - 라면을 냄비에 넣는다. - 라면이 다 끓었는가?(조건문) - 라면을 그릇에 담는다(출력) 이때 문자, 숫자, 이미지 등을 입력하고, 그에 대한 출력 값을 구하는 과정을 알고리즘이라 한다. 구조화란 문제를 분해하는 것을 말한다. 즉 반복적인 패턴을 단순화하는 과정을 의미하기도 한다. 추상화란 특정 사건에 대해서 모델링을 하는 과정을 말하며, 순서 절차를 도표로 만드는 것, 관례를 다이어그램으로 표현하는 것 등을 추상화라고 할 수 있다. 이러한 추상화를 위해서는 사건에 대한 분석과 설계가 우선 되어야 표현하기가 쉽다.

1.2 Training Contents and Teaching method

교육내용과 교수방법에는 Table 6과 같이 전체적으로 하이브리드 교수법을 이용한 완전학습이 되도록 설계하였다. 지식 습득단계에서 소프트웨어 기초이해, 코딩 표현 방법, 컴퓨팅 사고와 문제해결 과정을 이해 할 수 있는 이론 수업을 온라인과 오프라인을 활용한 플립러닝 교수법을 이용한다. 사고확장단계에서는 컴퓨팅 사고기반 문제해결 절차학습, 추상화/알고리즘을 통한 문제해결, 분해, 패턴인식, 프로그램 체험 수업을 오프라인의 실습/활동을 통해 컴퓨팅적 사고를 확장한다. 경험/성찰 단계에서는 문제를 찾고 해결하는 과정을 실습과 프로그램을 통해서 경험한다.

Table 6. Step-by-step Educational Contents / Teaching Method

Hybrid Teaching Method Step	Educational contents and teaching methods
Knowledge Acquizzational Step	<ul style="list-style-type: none"> <li>- Understanding software basics</li> <li>- Coding expression method</li> <li>- Problem analysis, data collection</li> <li>- Understanding computational Thinking and problem solving processes</li> </ul>
	On-Off Line, Flipped Learning
Thinking Expansion Step	<ul style="list-style-type: none"> <li>- Learning problem-solving procedures based on computing thinking</li> <li>- Problem solving with abstraction, algorithms</li> <li>- Problem solving with disassembly, pattern recognition</li> <li>- Problem finding and analysis, realize solution</li> </ul>
	Off Line, PBL(Problem Based Learning)
Experience/ Reflection Step	<ul style="list-style-type: none"> <li>- Experience a variety of programs</li> <li>- Practice entry program</li> <li>- Solving real-life problems through projects</li> </ul>
	Off Lene, PBL(Project Based Learning)

컴퓨팅 사고력을 키우는 수업을 연구하는 동시에 평가기준도 매우 중요하다. 기존의 평가방법을 가지고 창의성과 문제해결력을 평가하는 방법은 적절하지 않다. 아직 평가방법에 대한 많은 연구가 진행되지 않았으며, 앞으로 신중하게 연구되어야 할 부분이다.

IV. Conclusions

우리나라를 비롯한 많은 나라에서 초·중·등 교육에 SW교육이 시작되었으나 대학에서는 아직 일부 학교에 그치고 있다. 현재 소프트웨어 중심대학을 선정하여 운영하고 있고, 선정되지 않은 대학에서는 교양교육으로 운영하고 있지만 커리큘럼에서부터 어떤 기준으로 교육해야할지 표준안이 마련되어 있지 않고 있어 다양한 방법으로 운영되고 있었다.

여러 학교를 조사한 결과 소프트웨어 중심대학은 전공자를 비롯한 비전공자를 대상으로 코딩교육, SW융합교육, 프로그래밍 입문, 컴퓨팅 사고력, 알고리즘 등이 교과목으로 운영되고 있었다. 그러나 소프트웨어 중심학교는 국가과제로 운영되기 때문에 시설이나 환경이 지원되고 있으나 소프트웨어 중심학교가 아닌 경우에는 겨우 한두과목 개설하여 시험운영하고 기반 시설이나 환경 지원은 미비한 상황 이었다.

본 연구는 국내 대학의 SW교육 트렌드 및 사례조사와 함께 이를 위한 대학에서 SW교육 프로그램을 제시하고 있다는 것에 시사점이 있다고 볼 수 있다. 현재 SW교육을 대학에서 교양교육으로 활용가능하기 위해서 몇 가지 제언을 하고자 한다.

첫째, SW교육이 비전공자에게 이루어지기 위해서는 전공과 비전공의 융합하는 방법을 모색해야 한다. 인문, 예·체능, 자연 전공 학생들에게 무조건 프로그램을 교육하기는 어렵다. 비전공과 융합하여 다양한 교수학습 방법이 모색되어야 한다.

둘째, SW교육 운영에 있어 기관의 환경과 시설이 뒷받침되어야 한다. 기존의 융합교육도 교과목 개설과 더불어 환경이 뒷받침되지 않아 실패한 사례들을 보면 기존의 교육방법으로는 SW교육도 실패할 확률이 높다. 학교의 지원, 교수학습 방법 개발과 더불어 평가 방법까지 연구되어야 한다.

셋째, 융합교육에서 가장 큰 과제는 교수자의 융합적 사고가 미비하다는 지적을 가지고 있다. 이와 더불어 SW교육에서 전문적으로 연구한 교수자의 역할이 필요하다. 컴퓨팅 사고력을 위한 이론적 학습은 MOOC를 활용하여 온·오프라인의 수업으로 활용한다면 효과를 거둘 수 있으리라 생각한다.

대학에서 SW교육이 더 절실할 때이다. 왜냐하면 초·중·고에서 학습하기 위해서는 교사양성이 시급한 문제인데 대학에서 앞서 SW교육이 이루어져 컴퓨팅 사고력을 확보한다면 미래의 부모교육과 더불어 교사 양성까지 효과를 가져 올 수 있기 때문이다. 이러한 SW교육은 교수·학습 방법도 중요하지만 평가 방법 또한 기존의 방법과 많이 달라져야 한다. 창의·인재를 목표로 하고 있기 때문에 창의성, 문제해결력을 평가할 수 있는 다양한 방법들이 제시되어 한다. 이에 본 연구의 향후과제에서는 본 프로그램을 대학의 교양교육에 활용하고 구체적 목표와 나아갈 방향을 모색하는 것이다.

## REFERENCES

- [1] H. Y. Jung(2014), "An Empirical study on information liberal education in university based on IT fluency and computational thinking concept", Journal of the Korea Society of Computer and Information 19(2), pp. 263-274.
- [2] J. Y. Lim(2011), Present State of Programming Language Education and Suggestions for Its Improvement, Journal of practical engineering education 3(1), pp. 56-61.
- [3] G. S. Rue, S. H. Lee, J. W. Kim(2011), The Challenge of Korean University Education for the Training of IT Human Resources, SERI, Research Report.
- [4] J. S. Cha(2015), study of the field customized SW training course design based on the analysis of the field suitability of the university SW education, Smart Media Journal 4(4), pp. 86-92.
- [5] S. H. Park(2016), Study of SW Education in University to enhance Computational Thinking, Journal of Digital Convergence 14(4), pp. 1-10.
- [6] Wing, J. M. (2008), Computational thinking and thinking about computing, Philosophical Transactions of The Royal Society, 366, pp. 3717-3725.
- [7] Computational Thinking, <http://terms.naver.com>
- [8] CSTA, <http://www.csteachers.org/>
- [9] Exploring Computational Thinking, <https://www.google.com/edu/>
- [10] M. K. Park, G. H. Kim, T. Y. Kim(2014), The Effect of the RME-based Algorithmic Learning on Elementary Students' Problem Solving Ability for Improving Computational Thinking, Korean Journal of Teacher education. 30(4), pp. 179-193.
- [11] S. J. Jun(2017), The Effect of Design-Oriented Model (NDIS) based on Computational Thinking in SW Education, The Journal of Korean association of computer education, 20(2), pp. 13-21.
- [12] M. Y. Ryu, S. K. Han(2015), Development of Computational Thinking-based Educational Program for SW Education, Journal of The Korean Association of Information Education, 19(1), pp. 11-20.
- [13] M. S. Lee(2016), Hybrid teaching method utilizing MOOC in basic general education, Korean National Institute for General Education, Research Report.
- [14] Ministry of Education data, <http://www.moe.go.kr>.

## Authors



Myung Suk Lee received the B.S., M.S. and Ph.D. degrees in Computer Engineering from Keimyung University, Korea, in 2001, 2003 and 2009, respectively. Dr. Lee joined the faculty of the College of Liberal Education at

Keimyung University, Daegu, Korea, in 2013. She is currently a Professor in the Science and Technology at College of Liberal Education, Keimyung University. She is interested in Computer Network, Internet Ethics and Computer Education etc.