JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 2: 246~251, Feb. 2021

교양교육의 컴퓨팅사고력 수업이 문제해결능력에 미치는 영향

신좌철*

The Effects of Computational Thinking-based Liberal Education on Problem Solving Ability

ChwaCheol Shin*

*Assistant Professor, Department of Innovation and Convergence, Hoseo University, Chungcheong-namdo, 31499 Korea

요 약

본 연구는 컴퓨팅사고력 강좌의 문제해결능력을 파악하고 분석한 자료로 소프트웨어 교육 계획에 반영하기 위하여 분석되었다. 학습자를 대상으로 문제해결능력에 관한 사전·사후 설문을 진행하여 학습자의 어려운 점을 측정해, 소프트웨어 교육 방안을 모색하였다. 본 연구를 위하여 수업 주제와 내용 범위 및 활동 계획을 주당 2시간씩 15차시 분량의 소프트웨어 교육과정으로 구성하였고, 수강생 63명을 대상으로 설문을 하였다. 연구 결과는 소프트웨어 교육을 통한 문제 해결 능력 향상의 폭이 '이공계열 학과'보다도 '인문계열 학과'가 더 높게 나타났다. 본 연구 결과를 토대로, 창의 융합형 인재 양성을 위하여 사고나 인식을 근본적으로 규정하는 이론적 체계가 서로 융합하여야 한다. 또한, 향후 소프트웨어 교육이 다양한 분야의 비전공자 대상까지 확대되도록 개선해야 한다.

ABSTRACT

This study is analyzed survey based on classes of computational thinking to identify problem solving ability, to reflect them in curriculum plan for software education. Through analyzing difficulties on computational thinking learners by pre/post test on problem solving ability, the education method of software curriculum was proposed. For this study, the subject, scope of content, and activity plan were organized into 15 weeks of software curriculum for 2 hours per week, and questionnaire was conducted for 63 students. As a result, the 'Humanities Departments' have shown higher problem solving ability improvement than 'Science and Engineering Departments'. Based on the results, in order to cultivate creative fusion-type talent, the theoretical systems that fundamentally define thinking and perception must be fused with each other. In addition, software education should be improved to be extended to non-majors in various fields.

키워드 : 컴퓨팅적사고, 창의 역량, 교양교육, 문제해결능력, 소프트웨어교육

Keywords : Computational thinking, Creativity, Liberal arts education, Problem solving ability, Software education

Received 22 January 2021, **Revised** 26 January 2021, **Accepted** 5 February 2021

*** Corresponding Author** ChwaCheol Shin(E-mail:ccshin@hoseo.edu, Tel:+82-41-540-9658)
Assistant Professor, Department of Innovation and Convergence, Hoseo University, Chungcheong-namdo, 31499 Korea

Open Access http://doi.org/10.6109/jkiice.2021.25.2.246 print ISSN: 2234-4772 online ISSN: 2288-4165

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/ by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. 서 론

4차 산업 시대를 대비하는 측면에서 교육 현장에서는 사회적 요구에 발맞추어 소프트웨어 교육 모델을 수행해야 하는지에 대한 논의와 다양한 기술 발전이 이루어지면서 학문 간의 경계가 허물어지고 있는 상황이다. 따라서 인문학적 상상력과 과학적 창의력을 갖춘 인재들이 필요하며 컴퓨팅사고력(Computational Thinking)을 통하여 문제해결능력(problem solving ability)을 갖춘 인재 양성이 중요하다[1].

현재 많은 대학에서 과학기술정보통신부 주관으로 소프트웨어 중심대학 사업을 진행하면서 소프트웨어에 관련된 융합전공 교육과정이 있다. 그리고 대부분의 대학에서도 졸업 필수인 교양필수 교과로 프로그래밍 관련된 강좌를 많이 설치하고 있다. 그렇지만 많은 수강생들은 프로그래밍 교육의 중요성을 크게 느끼지 않거나 기피하는 현상이 있다[2].

이에 본 연구는 컴퓨팅사고력을 수강하는 학습자가 흥미를 지속적으로 가질 수 있도록 구성하였고 문제해결능력을 향상시키기 위해서 소프트웨어 관련 과목을 학습 영향에 미칠 수 있도록 구성하였다. 따라서 본 연구에서는 학습자를 상대로 문제해결능력에 대한 사전·사후 설문을 통하여 변화를 분석하고 학습자가 어려워하는 점이 무엇인지 조사하고 측정 하였다.

본 연구는 다음과 같이 내용으로 구성하였다. 서론에 이어서 2장에서는 관련 연구를 탐구하여 기술하였다. 3장에서는 연구 방법을 연구대상과 수업설계를 기술하였고, 4장에서는 연구 결과를 기술하였으며 마지막으로 결론 및 제언으로 맺는다.

Ⅱ. 관련 연구

2.1. 문제해결능력

문제는 주어진 환경과 상황 속에서 효과적으로 반응해서 해결해야 하는 상황으로 이런 문제는 현재 상황과 이루고자 하는 목표 상황 간의 차이가 인식될 때 개인에게 지각되는 것이다. 문제해결능력은 갈등 상황에 다양한 대안들을 생각해내고 대안 중에서 최선의 방법을 선택할 수 있는 일의 진행으로 부적응 상황을 해결하고자 하는 대처 방법의 하나로 최종 목표에 따른 현재 상황의 변화 시도를 위한 방법 수행이다. 문제해결능력은 직면한 문제 상황에 대해서 효과적인 대안을 발견하고 개입할 수 있는 수단을 형성하는 능력이다. 문제해결에 필요한 방법을 찾아내고 방법에 따른 결과를 비교하여 최적의 방법을 실행으로 옮기는 능력이며 달성하고자 하는 목표 간의 차이를 인식하여 이 차이를 해소시킬 수 있는 창의적 능력으로 소프트웨어 교육이 다양하게 정의되고 있다.

최근 들어 소프트웨어 교육에 대한 비중이 커지면서 이에 대한 능력을 갖추고 본인의 생각과 가치를 소프트웨어를 통해 구현과 문제를 해결할 수 있는 능력을 가진 인재 양성에 있다. 이와 같이 단순한 프로그래밍 능력의 비판적 사고와 문제해결능력을 동시에 갖추므로 대학이 지향하고자 하는 것은 소프트웨어 교육의 인재상과 같다. 소프트웨어 교육은 단순한 프로그래밍 교육이 아닌 문제해결능력까지 갖출 수 있는 인재상을 의미하는 핵심 요소이다.

현재, 컴퓨팅사고력 교육은 대학의 필수 교과로 소프트웨어 교육 과정이 확대 운영되어서 소프트웨어 중심 사회가 요구하는 컴퓨팅사고력 함양을 위하여 대학 교육에서 기초교육 모델이 되고[3] 소프트웨어 교육에 대한 중요성을 인식하여 한층 더 확대시켜 비전공자 대상으로 프로그래밍 교육을 진행하고 있다.

학습자가 프로그래밍을 어렵게 느끼는 것은 사고 영역 요소를 밝히고 컴퓨팅사고력의 관련성을 분석하여 관계있는 요소를 탐색한다. 본 연구를 통하여 학습자가 어려워하는 것은 문제를 이해 단계부터 코딩 단계까지 모두 어려워한다는 점을 알 수 있었다. 이런 상황에서 코딩을 한다면 학습자가 인지적 부담이 가중되기 때문에 사고에 대한 학습이 이루어져야 하고 이를 학습한 후에 프로그래밍을 학습한다면 인지적 부담이 감소되어 프로그래밍을 통하여 문제해결 학습에 집중할 수 있다[4]. 그러므로 본 연구에서는 소프트웨어 교육은 문제해결 중심의 형태로 설계하고 제시된 문제 상황에 맞는 해결 방법을 배우고 활용하여 스스로 해결할 수 있도록 한다[5].

2.2. 소프트웨어 교육

J. Wing(2008)은 컴퓨팅사고력이 컴퓨터 과학의 기초적인 개념을 익히는데 도움을 주고, 학습자가 갖추어야 할 기본 능력으로 자동화와 추상화를 통하여 문제해

결능력이라고 하였다. 컴퓨팅사고력은 주어진 문제에 대한 해결을 위하여 추상화 과정으로 문제에 대한 핵심 요소 추출과 모델링하고 컴퓨팅 기기를 활용하여 주어진 문제를 해결하는 자동화 능력을 의미한다고 하였다.

컴퓨터 과학 지식과 프로그래밍 능력이 우수하다고 하여 컴퓨팅사고 능력이 우수한 것은 아니다. 주어진 문제에 대한 인식과 분석, 자료 수집과 분석, 문제에 대한 해결 마련을 위해서 다양한 사고를 진행하고 이런 과정에서 컴퓨팅 능력을 활용하고 해결 방법을 정보과학적으로 구현하는 방법을 설계하는 과정이 컴퓨팅 사고이다.

4차 산업혁명 시대가 도래하면서 사회적으로 소프트웨어 교육에 많은 관심이 있다. 기존의 컴퓨터교육은 기본 소양과 컴퓨터 활용 등을 다루었다. 하지만 이제는 기존의 응용 소프트웨어 활용 위주의 컴퓨터 교육에서 벗어나 논리적 사고와 창의적인 문제해결능력을 향상시킬 수 있는 컴퓨터 교육으로 개발해서 현장에 적용 가능한 소프트웨어 교육으로 해야 한다[6].

최근에는 인공지능, 빅 데이터, 사물인터넷 등이 미래 산업에 중요성이 높아지면서 소프트웨어 교육이 인재양성 대안으로 제시되고 미래 사회를 설계하는데 필요한 문제해결능력을 함양하는 것이다[7].

본 연구에서는 소프트웨어 교육으로 4차 산업 혁명 시대의 미래 사회를 대비하고 컴퓨팅사고력을 갖춘 인재를 양성하기 위한 소프트웨어 교육에 초점을 맞추었다. 소프트웨어 교육의 목표로 언급되는 컴퓨팅사고력의 문제해결 과정으로 문제분해, 패턴인식, 추상화, 알고리즘과 자동화의 단계별 내용을 구체적으로 제시하였다. 대학에서 기초교양 필수로 소프트웨어를 가르치는 목적은 수강생이 미래 기술기반의 미래사회를 준비하도록 하기 위한 목표이다[8].

Ⅲ. 연구 방법

3.1. 연구대상

본 연구는 충청남도 A대학의 기초교양 교육에서 ‘AI와 컴퓨팅사고력’ 수업을 듣는 1학년 학생들을 대상으로 진행하였다. 이 실험을 위하여 연구대상은 2개 분반으로 인문계열 학과 수강생 31명과 이공계열 학과 수강생 32명이다.

3.2. 수업설계

본 연구 대상 강좌인 ‘AI와 컴퓨팅사고력’은 기초교양 교육으로 소프트웨어 교육 내용 범위를 표 1에 제시한 바와 같이 설정하였다.

Table. 1 Time-to-Time Learning and Strategy

Week	Subjects	Activities
1	Orientation	Understanding Changes in Society and the 4th Industrial Revolution
2	Understanding CT	• Definition of computing thinking and understanding of elements • Game analysis activity with CT
3	Computer structure	• Computer history • Computer structure and commands • Command drawing activity
4	Representation of data and bits	• Game with understanding binary and decimal • Various binary representation activities
5	Binary Representation and Multimedia	• Binary Expression and Compression Method of Multimedia • Image representation and compression activity
6	Decomposition	• Problem Solving and Coding Activities • Disassembly and application to major and real life problems
7	Pattern recognition	• Finding Methods and Examples • Pattern finder activity in various road shapes
8	Midterm Exam	
9	Abstraction	• Abstraction cases in various fields • Create a robot article sentence model
10	Algorithm representation	• Grid paper algorithms activities • The condition and expression of the algorithm
11	Selection structure	• Selection structure algorithm examples • Choose rescue card playing activities
12	Repeat structure	• Repeat structure algorithm • For loop dice game activity
13	Data structure	• Data structure in real life • Mud City (Minimum Height Tree) Activities
14	CT project	• CT element analysis activity in various games
15	Final Exam	

‘AI와 컴퓨팅사고력’ 강좌는 주당 2시간씩 15차시 분량 수업내용으로 진행하였고 2개 분반의 수강생을 대상으로 수업을 진행하였고 사전·사후 비교를 위해서 문제해결능력 검사를 실시하였다.

소프트웨어 교육이 미치는 영향과 상태를 확인하기 위하여 'AI와 컴퓨팅사고력' 강좌 교육 내용은 '컴퓨팅사고력 개념', '컴퓨터 기초', '멀티미디어 처리', '문제분해', '패턴인식', '추상화', '알고리즘', '자료 구조' 등으로 효과적인 학습 성취도를 파악하려 하였고 문제해결능력을 설문 조사로 실험하였다.

1주차에 각 분반에 대하여 문제해결능력 사전검사를 하였고, 전공에 따른 특성을 파악하면서 학습을 진행하고 15주차에 사후검사를 하였다.

Ⅳ. 연구내용 및 결과 분석

컴퓨팅사고력이 실험 집단과 통제 집단의 문제해결능력에 효과가 있는지 알아보기 위하여 인문계열과 이공계열 두 집단 간 동질성 검사와 사전·사후 문제해결능력 검사를 실시하였다. 본 연구를 위한 실험도구는 한국교육개발원에서 개발한 간편 창의적 문제해결력 검사지[9]를 사용해서 사전·사후 검사를 진행하였다.

4.1. 문제해결능력 검사 도구

본 연구의 문제해결능력 검사는 이석재 외(2003)가 개발한 검사 도구를 사용하였다. 문제해결능력 검사 도구는 전체 45문항으로 9개의 하위 영역별 5문항씩 구성되어 있다. 구성요소는 문제명료화, 원인분석, 대안개발, 계획/실행, 수행평가의 5개 요소로 구성되어 있고 Likert 5점 평정척도로 신뢰도가 검증된 검사 도구이다. 본 연구에서 문제해결능력의 내적신뢰도 계수인 Cronbach's α 값은 사전 0.873, 사후 0.778로 나왔다. 문제해결능력 영역과 문항 구성 및 내적 신뢰도는 표 2와 같다[9].

4.2. 두 집단 간 동질성 검사

소프트웨어교육의 문제해결능력에 대한 증진 방안을 제시하기 위하여 사전 검사 응답 결과를 독립표본 t-검정을 통하여 동질 집단인지 여부를 살펴보았다.

표 3에서 보는 것과 같이 두 집단 간의 차이는 문제인식(t=.336, p=.738), 정보수집(t=-1.193., p=.238), 분석(t=-.279, p=.782), 확산적사고(t=.991, p=.326), 의사결정(t=-.140, p=.890), 기획력(t=-.597, p=.553), 실행과 모험 감수(t=.960, p=.341), 평가(t=.610, p=.544), 피드백

Table. 2 Troubleshooting ability examination region, question configuration, and internal reliability

Region		Questi-on composi tion	Number of question	Cronbach's a	
Element of Aability	Subordinate region			Pre	Post
Problem clarification	Problem recognition	1 ~ 5	5	0.737	0.708
Cause analysis	Gathering information	6 ~ 10	5	0.607	0.648
	Analysis	11 ~ 15	5	0.746	0.635
Alternative development	Divergent thinking	16 ~ 20	5	0.751	0.652
	Decision making	21 ~ 25	5	0.666	0.678
Plan/ Action	Planning ability	26 ~ 30	5	0.618	0.654
	Action and risk-taking	31 ~ 35	5	0.670	0.602
Performance evaluation	Evaluation	36 ~ 40	5	0.625	0.589
	Feedback	41 ~ 45	5	0.711	0.590
Troubleshooting ability		1 ~ 45	45	0.873	0.778

(t=.008, p=.993)에서 모두 유의수준(p<.05)을 만족하지 못하였다. 따라서 두 집단 간의 차이는 통계적으로 유의미하지 않으므로 두 집단은 동질 집단이라고 볼 수 있다.

Table. 3 T-test for homogeneity of two groups

Categories	Science and Engineering (N=32)		Humanities (N=31)		t	p
	M	SD	M	SD		
1~5	3.775	.396	3.742	.384	.336	.738
6~10	3.425	.539	3.587	.539	-1.193.	.238
11~15	3.750	.463	3.787	.589	-.279	.782
16~20	3.344	.417	3.245	.373	.991	.326
21~25	3.725	.415	3.742	.542	-.140	.890
26~30	3.531	.688	3.626	.560	-.597	.553
31~35	3.488	.482	3.361	.560	.960	.341
36~40	3.800	.363	3.735	.472	.610	.544
41~45	3.769	.404	3.768	.537	.008	.993

4.3. 이공계열 학과의 실험결과

수강생의 문제해결능력 9가지 하위 영역에 어떠한 영향을 미치는지를 확인하기 위하여 두 집단에 각각 대응표본 t-검정을 실시하여 사전·사후 응답 차이를 분석

하였다.

표 4에서 보는 것과 같이 이공계열 학과에서는 문제인식(t=.390, p=.698), 정보수집(t=-.143, p=.887), 분석(t=.100, p=.921), 확산적사고(t=.150, p=.881), 의사결정(t=1.412, p=.163), 기획력(t=-.584, p=.561), 실행과 모험감수(t=-.225, p=.823), 평가(t=.381, p=.705), 피드백(t=-.486, p=.629)에서 모두 유의수준(p<.05)으로 통계적으로 유의미하지 않은 것으로 나타났으며 문제해결능력의 모든 하위 영역에서 사전 사후 변화가 없다는 것을 알 수 있었다. 이를 통하여 창의적인 소프트웨어 교육에 관한 새로운 모델의 필요성을 알 수 있다.

Table. 4 T-test for control group(Science and Engineering)

Categories	N=32	M	SD	t	p
1~5	pre-test	3.775	.396	.390	.698
	post-test	3.725	.607		
6~10	pre-test	3.425	.539	-.143	.887
	post-test	3.444	507		
11~15	pre-test	3.750	.463	.100	.921
	post-test	3.738	.534		
16~20	pre-test	3.344	.417	.150	.881
	post-test	3.325	.572		
21~25	pre-test	3.725	.415	1.412	.163
	post-test	3.563	.501		
26~30	pre-test	3.531	.688	-.584	.561
	post-test	3.625	.592		
31~35	pre-test	3.488	.482	-.225	.823
	post-test	3.519	.620		
36~40	pre-test	3.800	.363	.381	.705
	post-test	3.756	.539		
41~45	pre-test	3.700	.575	-.486	.629
	post-test	3.775	.658		

4.4. 인문계열 학과의 실험결과

표 5에서 보는 바와 같이 인문계열 학과에서는 문제인식(t=-2.323, p=.024), 정보수집(t=-2.111, p=.039), 분석(t=-2.632, p=.011), 확산적사고(t=-1.692, p=.096), 의사결정(t=-2.753, p=.008), 기획력(t=-2.357, p=.022), 실행과 모험 감수(t=-1.667, p=.101), 평가(t=-1.221, p=.227), 피드백(t=-1.472, p=.146)에서 확산적사고, 실행과 모험감수, 평가, 피드백 등 항목은 통계적으로 유의미하지 않은 것으로 나타났으며 문제해결능력의 모든 하위 영역에서 사전·사후 변화가 없다는 것을 알 수 있다. 그러나 문제인식, 정보 수집, 분석, 의사결정, 기획력 등을 묻는 항목에서는 그 차이가 유의미한 개선을 발견할 수 있었다. 이를 통하여 소프트웨어 융합교육의 필요성을 알 수 있다.

Table. 5 T-test for experience group(Humanities)

Categories	N=31	M	SD	t	p
1~5	pre-test	3.742	.384	-2.323	.024
	post-test	4.013	.524		
6~10	pre-test	3.587	.539	-2.111	.039
	post-test	3.884	.568		
11~15	pre-test	3.761	.525	-2.632	.011
	post-test	4.148	.628		
16~20	pre-test	3.245	.371	-1.692	.096
	post-test	3.523	.834		
21~25	pre-test	3.742	.542	-2.753	.008
	post-test	4.135	.583		
26~30	pre-test	3.626	.560	-2.357	.022
	post-test	4.013	.723		
31~35	pre-test	3.361	.560	-1.667	.101
	post-test	3.639	.738		
36~40	pre-test	3.781	.530	-1.221	.227
	post-test	3.948	.551		
41~45	pre-test	3.768	.537	-1.472	.146
	post-test	3.994	.664		

Ⅴ. 결론 및 제언

본 연구는 효과적인 소프트웨어 교육과정을 제시하고 새로운 교육 방안으로 가능성을 파악해보고자 하였다. 4차 산업 시대를 대비하기 위하여 대학에서도 전공자 뿐 아니라 비전공자를 대상으로 소프트웨어 교육의 중요성이 높아지고 있다. 이에 본 연구는 소프트웨어 교육을 효과적으로 운영하기 위한 방안을 모색하였다.

본 연구를 위하여 컴퓨팅사고력 수업 주제와 내용범위 및 활동 계획을 주당 2시간씩 15차시 분량의 소프트웨어 교육 과정 수업내용으로 개념 학습과 컴퓨팅 사고의 과제 및 실습으로 구성하였다.

본 연구는 충청남도 소재 A대학의 컴퓨팅사고력 수업을 이수한 63명의 수강생을 대상으로 설문을 실시하였다. 또한 컴퓨팅사고력 문제해결능력의 성취를 위한 소프트웨어 교육 과정을 개편하는 교육 내용으로 제안할 수 있었다.

연구 분석 결과로는 '이공계열 학과'보다도 '인문계열 학과'가 소프트웨어교육에 의한 문제해결능력 향상도가 높게 나타났다는 것을 알 수 있었으며, 창의적인 융합형 인재 양성에는 문제해결능력이 사고나 인식을 근본적으로 규정하는 이론적인 체계로써 서로 융합되는 것이 필요하다는 것을 알 수 있었다.

따라서 소프트웨어 교육이 학습자의 학습동기 유발과 효과적인 학습을 위하여 컴퓨팅 사고 수업을 창의적인 소프트웨어 교육 목표로 문제해결능력 성취와 흥미로운 교육 과정으로 도출하여야 한다.

향후 연구 과제로는 본 연구의 결과를 통하여 대학 교양교육 교과 운영을 다양하게 적용 확대하여 창의적인 소프트웨어 교육 모델을 새롭게 제안하고자 한다. 또한 수강생의 문제해결능력을 위한 개선 방안을 제시하고 프로그래밍 도구와 언어를 활용한 교양 교육을 다양한 과목들로 연구 대상을 확대하고자 한다.

ACKNOWLEDGEMENT

"This research was supported by the Academic Research fund of Hoseo University in 2019". (2019-0416)

REFERENCES

[1] S. M. Kim, K. S. You, K. C. Hong, and Y. B. Cho, "The Analysis of Resilience of Programming Class' Students for Basic Liberal Arts," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 23, no. 7, pp. 801-806, Jul. 2019.

[2] K. S. You, S. M. Kim, K. C. Hong, and S. Y. Choi, "The Analysis of Learning Demotivation according to Gender and Programming Subjects in Programming Class' Students of Liberal Arts," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 23, no. 6, pp. 704-710, Jun. 2019.

[3] S. H. Park, "Study of SW Education in University to enhance Computational Thinking," *Journal of Digital Convergence*, vol. 14, no. 4, pp. 1-10, 2016.

[4] K. S. Oh and S. G. Ahn, "A study on the relationship between difficulty in learning to program and Computational Thinking," *The Journal of Korean Association of Computer Education*, vol. 18, no. 5, pp. 55-62, 2015.

[5] Y. S. Lee, "Analyzing the effect of software education applying problem-solving learning," *Journal of Digital Convergence*, vol. 16, no. 3, pp. 95-100, 2018.

[6] K. K. Kim and J. Y. Lee, "Analysis of the Effectiveness of Computational Thinking-Based Programming Learning," *The Journal of Korean Association of Computer Education*, vol. 19, no. 1, pp. 27-39, 2016.

[7] A. H. Lee, "Domestic Research Trends Analysis of Software Education," *The Journal of Educational Information and Media*, vol. 24, no. 2, pp. 277-301, 2018.

[8] S. Y. Hong, J. Y. Seo, E. H. Goo, S. H. Shin, H. Y. Oh, and T. K. Lee, "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, Oct. 2019.

[9] J. M. Lee, "The Effects of Embedded-Based Programming Education on Problem Solving Ability, Self-Directed Learning Scale and Programming Ability of University Students," Master's Thesis, Ajou University, 2019.

신좌철(ChwaCheol Shin)

1990: 호서대학교 전자계산 이학사
1996: 호서대학교 전자계산 이학석사
2007: 호서대학교 컴퓨터공학 공학박사
현재: 호서대학교 혁신융합학부 조교수
※관심분야 : 소프트웨어교육, 컴퓨터융합교육