

파이썬 기반의 컴퓨터 프로그래밍 교육 모델 연구

강 환수¹ · 이종만² · 김희천^{3*}

¹동양미래대학교 컴퓨터정보공학과 교수

²동양미래대학교 경영정보학과 교수

³한국방송통신대학교 컴퓨터학과 교수

A Study on Computer Programming Education Model based on Python

Hwan-Soo Kang¹ · Jong-Man Lee² · Hee-Chern Kim^{3*}

¹Professor, Department of Computer Information Engineering, Dongyang Mirae University, Seoul, Korea

²Professor, Department of Management Information System, Dongyang Mirae University, Seoul, Korea

³Professor, Department of Computer Science, Korea National Open University, Seoul, Korea

[요 약]

국내의 대학에서 비전공자의 컴퓨팅 사고력 함양 목적으로 파이썬 언어가 많이 선택되고 있으나 전공자를 위한 프로그래밍 언어 교과목에서 파이썬의 선택은 적은 편이다. 본 연구에서는 컴퓨터 전공 학생 대상의 효과적 프로그래밍 교육을 위해, 전공자를 위한 파이썬 프로그래밍 학습역량을 도출하고 학습역량 기반의 파이썬 교육 모델 CS²를 제시하였다. 그리고 본 모델을 D 대학의 전공자를 대상으로 실제 교육에 적용하여 학생의 수업 전·후 역량을 대응표본 t-검정으로 비교 분석하였다. 분석 결과, 본 교육 모델 CS²의 사전·사후 검사 결과에서 유의미한 차이가 있었다. 본 연구의 교육 모델이 컴퓨터 전공자의 프로그래밍 교육에서 파이썬이 많이 활용되는 전환점이 되는 계기를 제공할 것으로 기대한다.

[Abstract]

Python is being chosen a lot as a programming language course to enhance computational thinking of non-computer majors in domestic universities. But for computer majors, Python's choice is uncommon. In this study, to effectively educate computer major students about programming, we derived the learning competencies factors of Python programming for majors and presented a Python education model - CS² based on learning abilities. We applied this model to the actual class of python programming for D-University computer majors, and analyzed data before and after class with a paired samples t-test. As a result of the analysis, there were significant differences between the pre and post-application of the CS² education model. We hope that the educational model of this study will provide a turning point when Python is widely used in programming education for computer majors.

색인어 : 파이썬 프로그래밍, 컴퓨터 프로그래밍 교육, 교육 모델, 학습역량

Key word : Python Programming, Computer Programming Education, Education Model, Learning Competencies

<http://dx.doi.org/10.9728/dcs.2020.21.4.693>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 06 March 2020; **Revised** 15 April 2020

Accepted 25 April 2020

***Corresponding Author; Hee-Chern Kim**

Tel: +82-2-3668-4657

E-mail: hckim@knou.ac.kr

I. 서론

한국을 비롯한 세계 각국은 인공지능과 빅데이터, 사물인터넷(IoT), 클라우드 컴퓨팅, 자율주행 등의 소프트웨어와 데이터를 기반으로 하는 4차 산업혁명에 맞아 인재양성을 위해 컴퓨터 프로그래밍 교육에 노력을 기울이고 있다. 4차 산업혁명 시대의 인재 역량은 단순 지식의 습득 능력보다 한 분야의 전문성을 바탕으로 컴퓨팅 사고력(computational thinking) 기반의 창의적인 융합 능력이 강조되고 있다. 1980년 시무어 페퍼트(Seymour Papert)가 처음 언급한 컴퓨팅 사고력은 2006년 자넷윙(Jeanette Wing)이 발표한 논문[1]에 의해 대중화되었다. 현재 초등학교를 비롯하여 중고등 과정에서 문제 해결 역량을 위한 컴퓨팅 사고력 증진 교육은 많이 연구[2]-[7]되고 있으며, 한국교육학술정보원은 초중고 학생의 SW 교육을 위해 5가지 형태의 교수·학습 모델을 제안하고 있다[8]. 대학 교육을 SW 중심으로 혁신하자는 SW 중점대학으로 선정된 많은 대학은 신입생 전체를 대상으로 필수 교양 과목으로 컴퓨팅 사고력, SW 융합 교육, 프로그래밍입문 등을 개설·운영하고 있다[9]-[17]. 특히 파이썬은 문법적으로 매우 간단해 쉽게 배울 수 있어 비전공자를 위한 교육에 많이 활용되고 있다[18]-[22]. 최근 연구 중에는 이 5가지 교수·학습 모델 중에서 시연 중심의 모델을 데이터 분석용 R 언어에 적용하여 데이터 분석 기반 SW 교육의 교수·학습 방법을 제안하고 있다[23]. 파이썬은 웹서버 구축에서 인기를 얻기 시작해 데이터 과학, 머신러닝·딥러닝 그리고 범용 소프트웨어 개발에 이르기까지 학계와 업계에서 중요한 프로그래밍 언어로 자리매김하고 있다. 파이썬은 다양한 학문에 적용되는 사례가 늘고 있으며, 수학 교육에도 파이썬 기반의 코딩 교육이 활용되고 있다[24]-[25]. 그런데도 아직 컴퓨터 관련 전공 학과에서는 프로그래밍 교육에 관한 큰 변화가 없으며 파이썬 교육 사례나 연구도 찾아보기 힘든 것이 현실이다. SW 전공자를 위한 프로그래밍 언어에 C 언어 교육에 블록 기반의 교육용 프로그래밍 언어(EPL)인 스크래치(Scratch)를 활용하는 수업 모형 설계 사례 정도이다[26]. 4차 산업혁명은 많은 산업이 정보기술과 융합하여 지능 디지털 기술로 변환(intelligent digital technology transformation)되는 혁명 시대로 컴퓨팅 사고력 기반의 소프트웨어 교육과 컴퓨터 전공자의 프로그래밍 교육의 중요성이 매우 커지고 있다. 컴퓨터 공학은 바로 4차 산업혁명의 인재양성을 위한 핵심 학과다. 특히 컴퓨터 관련 전공자에게 해결하고자 하는 문제를 명확히 이해하고 컴퓨터가 효과적으로 문제를 풀어 문제를 해결하는 과정을 학습하는 프로그래밍 교육은 무엇보다도 중요하다.

본 연구는 컴퓨터공학 전공자에게 효과적인 컴퓨터 프로그래밍 교육을 위해 대학 컴퓨터 전공자에게 필요한 파이썬 기반의 교육 목표를 연구하고 목표를 지향하는 학생 역량을 도출하여 프로그래밍 교육 모델을 제안한다. 제안된 교육 모델을 직접 수업에 운영하여 교육의 유의미한 효과를 검증하고자 한다.

II. 파이썬 기반 컴퓨터 프로그래밍 교육 모델

2-1 컴퓨터 프로그래밍 교육 모델: CS²

파이썬의 문법은 매우 쉽지만, 확장성이 뛰어나 웹 프로그래밍 분야에서 데이터 처리와 데이터 시각화 분야, 그리고 머신러닝·딥러닝 분야까지 광범위하게 활용될 수 있다. 현재 컴퓨터 전공의 프로그래밍 교육 언어로 많이 활용되고 있는 언어는 자바와 C, C++가 주류이다. 파이썬이 쉽다는 장점이 있어 비전공자는 처음 배우는 언어가 주로 파이썬이다. 전공자를 위한 본 컴퓨터 프로그래밍 교육 모델에서는 자바나 C를 먼저 이수한 학생을 대상으로 파이썬 기반 프로그래밍 교육을 제안한다. 컴퓨터 전공 교과에서 자바나 C를 제외할 수는 없으며 파이썬이 4차 산업혁명의 핵심 기술인 데이터과학과 머신러닝·딥러닝 등의 분야로 확장하는 데 적합한 언어이므로 자바나 C의 선행학습을 전제로 교육 모델을 설계한다.

본 연구에서 제안하는 파이썬 기반 컴퓨터 프로그래밍 교육 모델은 또 다른 새로운 언어의 효과적인 습득을 위해 프로그래밍 언어를 전반적으로 정리·이해하고 파이썬 언어의 장점을 살려 데이터과학과 인공지능 등의 다양한 분야로 확장하여 문제 해결을 준비하는 관점에서 컴퓨터 프로그래밍 교육 모델을 제안한다. 본 연구의 컴퓨터 프로그래밍 교육 모델에서 새로운 언어의 효과적 습득과 다양한 분야로 확장 패러다임을 기반으로 프로그래밍 언어의 이해와 학생이 배워야 할 학습역량을 첫 번째, 프로그래밍 언어의 전반적 이해, 두 번째 파이썬 구조와 문법, 그리고 마지막 세 번째는 모듈과 패키지를 활용한 기능 확장으로 설정하였다. 다음 그림 1은 본 연구에서 제안하는 컴퓨터 프로그래밍 교육 모델이다. 세 가지 학습 목표인 COP, SSP, SMP를 토대로 CS²라 명명하였다.

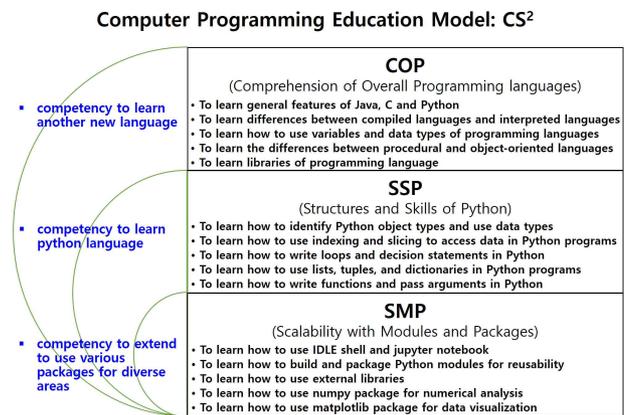


그림 1. 컴퓨터 프로그래밍 교육 모델: CS²
Fig. 1. Computer programming education model: CS²

전공자에게 향후 사용될 새로운 언어의 빠른 적응과 습득을 위해 프로그래밍 언어에 대한 전반적 이해(COP)가 필수적이다. 프로그래밍 언어의 전반적 이해 목표는 컴파일러와 인터프

리터 언어의 이해, 객체지향과 절차적 프로그래밍 방식의 이해, 변수와 자료형의 필요성, 라이브러리의 활용으로 구성된다. 컴퓨터 전공자에게 프로그래밍 언어 교육의 핵심은 선정된 프로그래밍 언어로 해결하고자 하는 문제를 구현하는 것이다. 파이썬 구조와 문법(SSP) 목표는 바로 파이썬을 활용한 구현 능력이다. 즉 파이썬 언어의 개발환경을 익히고, 자료형과 변수, 제어구조, 자료구조를 학습하는 것이다. 마지막으로 모듈과 패키지를 활용한 기능 확장(SMP) 목표는 파이썬의 가장 큰 장점인 4차 산업혁명의 화두인 머신러닝·딥러닝 등의 다양한 분야에 적합하다는 것을 준비하는 활동 과정이다. 파이썬은 데이터 과학 및 딥러닝 알고리즘 생성에 널리 활용되는 범용 고급 프로그래밍 언어이다. 본 역량은 통해 numpy와 matplotlib를 체험하고 향후 Theano, TensorFlow, Keras와 같은 인공지능 프레임워크 활용을 준비하는 데 필요하다.

본 교육 모델에서 제안한 세 가지 교육 목표 COP, SSP, SMP에서 컴퓨터 프로그래밍 교육 모델을 CS²라 칭하며, 이 수업을 수강하는 학생이 배워야 할 구체적인 학습역량은 다음과 같다.

1) 프로그래밍 언어의 전반적 이해: COP

본 교육 모델을 수강하는 학생은 이미 자바를 두 학기 동안, C를 한 학기 배운 학생으로 프로그래밍 언어의 전반적 이해를 목표로 한다. 본 학습역량에서는 파이썬을 자바와 C 언어와 비교하여 특징을 이해하고, 컴파일러 언어와 인터프리터 언어를 구분할 수 있어야 한다. 또한, 자료형과 변수를 이해하고 활용할 수 있어야 하며, 절차적 언어와 객체지향 언어의 차이를 이해하고, 프로그래밍 언어의 라이브러리를 활용할 수 있어야 한다.

2) 파이썬 구조와 문법: SSP

프로그래밍 언어의 학습 요소는 기본적으로 자료형과 변수, 제어구조, 자료구조 그리고 개발환경과 도구라고 볼 수 있다. 본 학습역량은 파이썬의 학습 요소를 기반으로 파이썬의 다양한 자료형이 모두 클래스라는 것을 이해하고 활용할 수 있는 파이썬 문법의 특징과 요소를 포함한다. 또한 파이썬의 시퀀스(sequences)를 이해하고 첨자와 슬라이싱(slicing)으로 참조할 수 있으며 파이썬의 조건문과 반복문을 이해하고 활용할 수 있어야 한다. 그리고 파이썬의 대표적 자료구조인 리스트와 딕셔너리를 이해하고 활용할 수 있어야 한다. 마지막으로 파이썬의 함수를 이해하고 직접 함수를 만들어 활용하거나 내장 함수를 사용할 수 있어야 한다.

3) 모듈과 패키지를 활용한 기능 확장: SMP

본 학습역량은 파이썬의 가장 큰 장점인 기능 확장 능력이다. 학생은 다양한 분야에 적합한 파이썬 개발환경을 이해하며 모듈과 패키지를 활용할 수 있어야 한다. 또한, 서드 파티(third party) 라이브러리를 이해하고 외부 라이브러리를 활용할 수 있으며, 대표적인 수리 분야의 패키지인 numpy와 시각화 패키지인 matplotlib를 활용할 수 있어야 한다.

2-2 프로그래밍 교육 모델 주간 일정

본 연구에서 제안한 컴퓨터 프로그래밍 교육 모델 CS²는 서울 소재 D 대학 2019학년도 2학기, 컴퓨터정보공학과와 두 개 반에서 운영되었다. 파이썬프로그래밍 교육 일정에서, 제안한 학습역량 COP, SPP, SMP의 관계와 학습 내용은 표 1과 같다. 수업은 주간 3시간의 이론과 실습이며, 중간·기말시험 2주를 포함하여 총 15주 일정으로 운영하였다.

표 1. 교육 모델 CS²의 주간 일정
Table 1. Week schedule of education model CS²

Week	Subject related	Contents
1	COP	Python and programming language overview
2	COP	Interpreted and compiled programming language IDLE shell and various python development environment
3	COP	Various data types and variables, IO function Anaconda and jupyter notebook
4	SSP	Conditional structure
5	SSP	Iteration structure
6	SSP	List and tuple
7	SSP	Dictionary and set
8	test	Mid test
9	SSP	User defined and built-in function
10	SMP	Module and package
11	SMP	Numpy and matplotlib package
12	SMP	GUI programming of tkinter and pygame
13	SSP	Exception handling and file input output
14	COP	Object-oriented programming language overview Classes and Objects in python
15	test	Final test

III. 모델 검정 방법

3-1 연구 대상

본 연구에서 제안한 컴퓨터 프로그래밍 교육 모델 CS²의 학생 역량 효과 검증을 위해 2년제 D 대학의 전공자를 대상으로 한 파이썬프로그래밍 교과목에서 설문을 시행했다. 파이썬프로그래밍 수업은 2019년 2학기, 2학년 과정으로 각각 38명과 30명 2반으로 운영되었다. 교육 모델 운영의 대응표본 t-검정을 위해 사전·사후 검사 모두 참여한 학생은 총 46명이었다.

3-2 연구 도구

교육 효과 검증을 위한 설문 문항은 본 교육 모델의 학생 역량인 ‘프로그래밍 언어의 전반적 이해’에 대해 5 문항, ‘파이썬 구조와 문법’ 5 문항, 그리고 ‘모듈과 패키지를 활용한 기능 확장’ 5 문항, 총 15문항으로 구성되어 있다. 각 문항은 ‘전혀 그렇지 않다’의 1점, ‘매우 그렇다’의 5점인 리커트 5점 척도로 평정하였다. 학생의 학습역량인 설문 문항의 주제와 내용은 표 2와

같다.

표 2. 교육 모델 CS²의 효과 검증을 위한 설문 문항

Table 2. Questionnaires to validate the effectiveness of education model CS²

Subject area	Item	Item contents
COP (Comprehension of Overall Programming languages)	COP-1	Features of Java, C and Python
	COP-2	Distinguishing compiler languages from interpreter languages
	COP-3	Understanding variables and data types of programming languages
	COP-4	Understanding the differences between procedural and object-oriented languages
	COP-5	Understanding libraries of programming language
SSP (Structures and Skills of Python)	SSP-1	Understanding data types and classes
	SSP-2	Accessing and slicing of python sequences
	SSP-3	Practical usage of control structures
	SSP-4	Practical usage of data structures
	SSP-5	Practical usage of functions
SMP (Scalability with Modules and Packages)	SMP-1	Python's various development tools
	SMP-2	Understanding modules and packages
	SMP-3	Utilization of external libraries
	SMP-4	Practical usage of numpy package
	SMP-5	Practical usage of matplotlib package

3-3 설문 문항의 정규성과 신뢰도

본 연구의 사전문항 정규성 검증을 위한 샤피로-윌크(Shapiro-Wilk) 검정과 왜도(skewness) 및 첨도(kurtosis) 검사 결과는 표 3과 같다. 분석 결과 유의확률이 대부분 .0으로 유의수준 $p > 0.05$ 를 만족하지 못하나 왜도와 첨도는 -2와 2 사이의 값으로 대칭성을 만족하고 뾰족한 정도를 크게 벗어나지 않은 것으로 검증되었다.

표 3. 정규성 검정

Table 3. Normality Test

Item	Kolmogorov-Smirnov			Shapiro-Wilk			Skewness	Kurtosis
	Statistic	df	Sig.	Statistic	df	Sig.		
COP-1	.301	46	.000	.850	46	.000	.043	.969
COP-2	.263	46	.000	.881	46	.000	.086	.457
COP-3	.205	46	.000	.882	46	.000	.050	-.880
COP-4	.227	46	.000	.891	46	.000	.160	-.248
COP-5	.244	46	.000	.891	46	.000	-.176	.197
SSP-1	.225	46	.000	.902	46	.001	-.124	-.361
SSP-2	.220	46	.000	.887	46	.000	.495	-.141
SSP-3	.188	46	.000	.861	46	.000	-.057	-1.323
SSP-4	.196	46	.000	.860	46	.000	.558	-.074
SSP-5	.227	46	.000	.868	46	.000	.012	-.964
SMP-1	.277	46	.000	.862	46	.000	.401	-.401
SMP-2	.224	46	.000	.862	46	.000	.166	-.693
SMP-3	.233	46	.000	.829	46	.000	.352	-.983
SMP-4	.252	46	.000	.824	46	.000	.604	-.632
SMP-5	.252	46	.000	.824	46	.000	.604	-.632

연구가설 검증에 앞서 설문 문항의 신뢰성을 검증하였다. 신뢰성이란 측정 도구가 동일한 개념에 대해 측정을 반복했을 때 같은 측정값을 얻을 가능성이 있다. 본 연구에서 신뢰성은 크론바흐 알파(Cronbach's Alpha) 계수로 확인하였다. 0~1 값을 가지는 크론바흐 알파 계수는 일반적으로 0.6 이상이면 신뢰성이 있는 것으로 간주한다. 설문 문항에 대한 신뢰도 통계량은 표 4와 같다.

표 4. 신뢰도 통계량

Table 4. Reliability Statistics

Cronbach's Alpha		N of Items
Pre	Post	
.930	.973	15

실험결과 설문 문항 전체에서 검정 사전-사후의 크론바흐 알파 계수는 각각 .930과 .973으로 모두 0.9 이상으로 매우 높은 신뢰도가 나타났다. 다음 표 5는 사전 설문 항목 각각의 신뢰도를 보여주는 항목 총계 통계량이다.

표 5. 설문 문항의 항목 총계 통계량

Table 5. Questionnaires Item-Total Statistics

Subject area	Item	Scale Mean if Item Deleted	Scale Variance if Item Deleted	Corrected Item-Total Correlation	Cronbach's Alpha if Item Deleted
COP	COP-1	34.74	87.308	.578	.927
	COP-2	34.87	87.094	.589	.927
	COP-3	34.37	87.705	.488	.930
	COP-4	34.80	87.228	.570	.928
	COP-5	34.74	85.175	.695	.924
SSP	SSP-1	35.00	84.933	.660	.925
	SSP-2	35.54	83.054	.729	.923
	SSP-3	35.30	82.394	.683	.925
	SSP-4	35.70	82.128	.785	.921
	SSP-5	35.52	83.855	.735	.923
SMP	SMP-1	35.54	85.587	.672	.925
	SMP-2	35.67	85.069	.747	.923
	SMP-3	35.89	84.943	.745	.923
	SMP-4	35.93	86.240	.619	.926
	SMP-5	35.93	86.240	.619	.926

표 5의 다섯 번째 열인 '수정된 항목-전체 상관관계(Corrected Item-Total Correlation)'는 신뢰도가 낮은 항목을 알려주는 지표인데, 결핍값은 모두 0.3을 초과한다. 보통 0.3 미만이면 신뢰도를 저해하는 항목으로 판단한다. 표 5 마지막 열인 '문항이 삭제된 경우 크론바흐 알파(Cronbach's Alpha if Item Deleted)'는 해당 하나의 문항이 삭제되었다고 가정했을 때, 전체의 신뢰도 계수가 어떻게 변하는지 알려주는 값이다. 따라서 어떠한 문항을 지웠다는 가정하에 전체 문항의 신뢰도 값이 .930보다 높게 나타나면 그 문항은 제거 대상이 될 수 있다. 실험결과 이 값이 .930보다 높게 나타난 문항은 없다.

IV. 교육 모델 CS² 연구 결과

4-1 프로그래밍 언어의 전반적 이해 역량 결과

본 교육 모델에서 제시한 프로그래밍 언어의 전반적 이해 역량의 효과성을 파악하기 위해 사전-사후 조사를 시행한 결과, 참여자의 사전-사후 조사 결과의 통계량은 다음 표 6과 같다. 자바와 C 언어를 배웠기 때문에 대부분의 문항에서 사전 결과가 높으며, 특히 프로그래밍 언어의 가장 기본인 변수와 자료형에 대한 이해가 사전-사후 모두 높다는 것을 알 수 있다.

표 6. 프로그래밍 언어의 전반적 이해(COP)의 대응표본 통계량
Table 6. Paired samples statistics of COP

Item	Pre			Post		
	Mean	Std. dev	Std. e mean	Mean	Std. dev	Std. e mean
COP-1	3.09	.865	.128	3.85	.759	.112
COP-2	2.96	.868	.128	3.85	.942	.139
COP-3	3.46	.959	.141	4.02	.856	.126
COP-4	3.02	.882	.130	3.76	.874	.129
COP-5	3.09	.890	.131	3.72	.958	.141

프로그래밍 언어의 전반적 이해 역량에 대한 사전-사후검사 차이에 대한 대응표본 t-검정 결과는 표 7과 같다. 검증 결과, 모든 역량에 있어 사전-사후검사 간에 $p < .05$ 수준에서 통계적으로 유의미한 차이가 있었다. 결과로 본 교육 모델을 적용한 수업이 학생들의 프로그래밍 언어의 전반적 이해 역량이 향상되었음을 알 수 있다. 특히 사전-사후 차이가 가장 큰 COP-2의 결과($t = -6.716, p = .000$)는 인터프리터 언어인 파이썬을 개발환경 IDLE 셸과 주피터 노트북으로 학습하면서 인터프리터 언어와 컴파일러 언어와의 차이를 인식했을 것으로 파악된다.

표 7. 프로그래밍 언어의 전반적 이해(COP)의 대응표본 t-검정 결과

Table 7. Paired samples t-test of COP

Item	Paired Diff.			t	p
	Mean	Std. dev	Std. e mean		
COP-1	-.761	.947	.140	-5.448	.000
COP-2	-.891	.900	.133	-6.716	.000
COP-3	-.565	.958	.141	-4.001	.000
COP-4	-.739	.905	.133	-5.538	.000
COP-5	-.630	1.019	.150	-4.196	.000

4-2 파이썬 구조와 기능 역량 평가

본 교육 모델의 파이썬 구조와 기능 역량의 효과성을 파악하기 위해 사전-사후 조사를 시행한 결과, 참여자의 사전-사후 조사 결과의 통계량은 다음 표 8과 같다. 문항 SSP-3의 사후 조사(mean = 4.15)에서 파이썬의 조건과 반복인 제어구조의 이해가 높다는 것을 알 수 있다.

표 8. 파이썬 구조와 기능(SSP)의 대응표본 통계량

Table 8. Paired samples statistics of SSP

Item	Pre			Post		
	Mean	Std. dev	Std. e mean	Mean	Std. dev	Std. e mean
SSP-1	2.83	.950	.140	3.87	.909	.134
SSP-2	2.28	1.004	.148	3.72	.935	.138
SSP-3	2.52	1.110	.164	4.15	.894	.132
SSP-4	2.13	1.002	.148	3.85	.918	.135
SSP-5	2.30	.940	.139	3.96	.842	.124

파이썬 구조와 기능 역량에 대한 사전검사와 사후검사 차이에 대한 대응표본 t-검정 결과는 표 9와 같다.

표 9. 파이썬 구조와 기능(SSP)의 대응표본 t-검정 결과

Table 9. Paired samples t-test of SSP

Item	Paired Diff.			t	p
	Mean	Std. dev	Std. e mean		
SSP-1	-1.043	.988	.146	-7.164	.000
SSP-2	-1.435	1.167	.172	-8.337	.000
SSP-3	-1.630	1.306	.193	-8.469	.000
SSP-4	-1.717	1.344	.198	-8.664	.000
SSP-5	-1.652	1.140	.168	-9.833	.000

검증 결과, 모든 역량에 있어 사전-사후검사 간 통계적으로 유의미한 차이가 있었다. 결과로 본 교육 모델을 적용한 수업으로 학생이 파이썬 구조와 기능 역량이 향상되었음을 알 수 있다. 대부분 사전-사후 간 차이는 1.4 이상으로 명확하게 나타났다. 또한, 파이썬의 특징적 자료구조인 리스트와 딕셔너리의 역량 문항인 SSP-4의 교육 효과(mean = -1.717)가 가장 크고, 파이썬 자료형이 클래스라는 내용인 SSP-1이 상대적으로 차이가 다소 작음(mean = -1.043)을 알 수 있다. 이는 파이썬이 객체지향 언어이나 객체지향에 관한 수업 시간이 3시간 정도로 상대적으로 매우 적은 결과라고 추정된다. 이에 대한 보완은 필요할 것으로 보인다.

4-3 모듈과 패키지를 활용한 기능 확장 역량 평가

본 교육 모델의 모듈과 패키지를 활용한 기능 확장 역량의 효과성을 파악하기 위해 사전-사후검사를 시행한 결과, 참여자의 결과 통계량은 다음 표 10과 같다.

표 10. 모듈과 패키지를 활용한 기능 확장(SMP)의 대응표본 통계량

Table 10. Paired samples statistics of SMP

Item	Pre			Post		
	Mean	Std. dev	Std. e mean	Mean	Std. dev	Std. e mean
SMP-1	2.28	.886	.131	3.70	.866	.128
SMP-2	2.15	.842	.124	3.67	.896	.132
SMP-3	1.93	.854	.126	3.50	.888	.131
SMP-4	1.89	.900	.133	3.59	.956	.141
SMP-5	1.89	.900	.133	3.50	.937	.138

검사 결과, 모듈과 패키지의 개념이 상대적으로 다른 역량보

다 학생에게는 다소 어렵게 생각된다는 시사점을 알 수 있다. 향후 모듈과 패키지를 활용한 기능 확장 역량에 대한 보완이 필요할 것으로 보인다.

파이썬 모듈과 패키지를 활용한 기능 확장 역량에 대한 사전-사후검사 차이에 대한 대응표본 t-검정 결과는 표 11과 같다. 검증 결과, 모든 역량에 있어 사전-사후검사 간 통계적으로 유의미한 차이가 있었다. 이 결과로 본 교육 모델을 적용한 수업으로 학생이 파이썬 모듈과 패키지를 활용한 기능 확장 역량이 향상되었음을 알 수 있다.

표 11. 모듈과 패키지를 활용한 기능 확장(SMP)의 대응표본 t-검정 결과

Table 11. Paired samples t-test of SMP

Item	Paired Diff.			t	p
	Mean	Std. dev	Std. e mean		
SMP-1	-1.413	1.087	.160	-8.818	.000
SMP-2	-1.522	1.090	.161	-9.468	.000
SMP-3	-1.565	1.148	.169	-9.247	.000
SMP-4	-1.696	1.152	.170	-9.981	.000
SMP-5	-1.609	1.085	.160	-10.058	.000

V. 결론

인공지능과 빅데이터로 대표되는 4차 산업혁명 시대에 파이썬 언어의 인기는 대단하다. 중고 학생과 대학의 비전공자나 교양 교과목에서의 코딩교육은 학습자의 문제해결 능력, 논리 사고력, 창의성을 향상 등에 그 목적이 있다. 파이썬은 구문이 매우 간단하고 문법도 쉽게 배울 수 있어 비전공자의 코딩교육에 매우 적합하다. 파이썬은 초보자에게 적합한 언어이면서도 확장성도 매우 뛰어나 웹 프로그래밍 분야에서 데이터 처리와 시각화 분야, 인공지능의 머신러닝, 딥러닝 분야까지 광범위하게 활용되고 있다. 그런데도 현재 컴퓨터 전공자에게 프로그래밍 교육 언어로 파이썬을 가르치는 사례는 적은 편이다. 많은 대학의 컴퓨터 전공 학과에서는 자바나 C, C++ 언어가 주로 활용된다. 본 연구는 컴퓨터 전공에서 파이썬 교육이 미흡한 상황에서 전공자에게 적합한 파이썬 기반 프로그래밍 교육 모델 CS²를 제안하고 본 제안 교육 모델이 프로그래밍 학습역량에 미치는 영향을 확인하기 위해 수행되었다. 연구 결과, 프로그래밍 언어의 전반적 이해(COP), 파이썬의 구조와 기능(SSP), 모듈과 패키지를 활용한 기능 확장(SMP) 부분의 학생 역량이 통계적으로 유의미한 증가가 있었다.

본 연구의 교육 대상자는 자바나 C 언어를 배운 컴퓨터 전공 학생이며, 졸업 후 새로운 프로그래밍 언어 학습에서 필요한 역량 향상과 데이터과학과 딥러닝 등의 다양한 분야로의 준비를 위한 교육 목표를 분석하고 학생들이 배워야 할 학습역량을 규정하였다. 그 결과, 파이썬 기반 컴퓨터 프로그래밍 교육 모델 CS²는 4차 산업혁명 패러다임이 요구하는 컴퓨터 전공자 교육을 위해 프로그래밍 언어의 전반적 이해, 파이썬의 구조와 기능, 모듈과 패키지를 활용한 기능 확장이라는 학생 역량 향상에

목적들을 두었다. 제안한 프로그래밍 교육 모델 CS²의 수업 운영 전과 후 학생 역량이 변화를 검증하기 위해 파이썬 프로그래밍 수업에서 전공자 46명의 설문 검사로 대응표본 t-검정을 시행하였다. 첫 번째, 프로그래밍 언어의 전반적 이해 역량에 검증 결과에서 학생 역량이 향상되었음을 알 수 있었다. 특히 개발환경 IDLE 쉘과 주피터 노트북으로 인터프리터 언어인 파이썬을 학습하면서 인터프리터 언어와 컴파일러 언어와의 차이를 명확히 이해했다는 점도 파악할 수 있었다. 두 번째, 파이썬 구조와 기능 역량을 검증한 결과에서도 통계적으로 유의미한 차이가 있었다. 전반적으로 파이썬 자체 프로그래밍 이해를 묻는 파이썬 구조와 기능 역량에 대한 설문에서 학생 역량 향상에 유의미 정도가 컸으며 특히, 파이썬의 특징적 자료구조인 리스트와 딕셔너리의 역량이 크게 향상되었음을 알 수 있었다. 세 번째 학생 역량 검증 결과, 본 교육 모델을 적용한 수업으로 학생의 파이썬 모듈과 패키지를 활용한 기능 확장 역량이 향상되었음을 알 수 있었다.

본 연구에서 제안한 교육 모델을 수업에 적용한 연구 결과에서 의의와 시사점을 살펴보면, 전공자에게 파이썬 프로그래밍 교육은 파이썬 자체의 코딩 역량을 학습하는 효과는 물론이며, 이전에 배운 자바나 C 프로그래밍 언어를 복습하고 정리하며, 졸업 후 다시 새로운 프로그래밍 언어의 효과적인 습득을 준비하는 차원에서 유의미한 교육의 성과가 있었다. 다만 데이터시각화 등의 관심 분야의 확장과 파이썬의 객체지향 프로그래밍을 심도 있게 다루기에는 일주 3시간의 수업 시간이 다소 부족하다고 생각된다. 향후 이러한 점을 보완하고자 수업 시간의 배분이나 확대와 함께 새로운 교과목의 개설도 심도 있게 연구하고자 한다. 본 연구에서 제안한 교육 모델을 계기로 컴퓨터 전공자의 프로그래밍 교육에서 4차 산업혁명에 적합한 파이썬 프로그래밍 언어가 많이 활용되는 전환점이 되기를 기대한다.

감사의 글

본 연구는 2019년도 동양미래대학교의 학술연구비 지원으로 이루어진 연구로서, 관계부처에 감사드립니다.

참고문헌

- [1] Wing, J. M, "Computational Thinking", *Communication of the ACM*, Vol. 49, No. 3, pp. 33-35, 2006.
- [2] H, C. Lee, H. S. Lee, J. S. Sung, S. W. Jung, S. H. Kim, S. H. Kim, J, H, Kim, J, Y, Park, and B. M. Lee, A study on surveying the actual conditions and evaluating the effectiveness of SW education in elementary and secondary schools, Ministry of Science, ICT and Future Planing, 15-JINHEUNG-077, 2015.
- [3] E. J. Kim, "The Effect of Convergence Lesson Plan and Teaching Demonstration for Enhancing Creative

- Competency of The Pre-service Teachers,” *Journal of The Korea Contents Association*, Vol. 19 No. 3, pp. 466-474, 2019.
- [4] I. H. Yoo, “A Study on SW Development Process for Increasing Computational Thinking,” *KIPS Tr. Software and Data Eng.*, Vol. 5, No. 2 pp. 51-58, 2016.
- [5] Y. J. Ju and D. S. Ma, “The Development of Abstractable Competency Assessment Standards for the Measurement of Computational Thinking,” *Journal of The Korean Association of Information Education*, Vol. 22, No. 3, pp. 375-383, 2018
- [6] J. C. Oh and J. H. Kim, “Predicting the Effect of Puzzle-based Computer Science Education Program for Improving Computational Thinking,” *Journal of The Korean Association of Information Education*, Vol. 23, No. 5, pp. 499-511, 2019.
- [7] J. M. Lee and E. J. Ko, “The Effect of Software Education on Middle School Students’ Computational,” *Journal of The Korea Contents Association*, Vol. 18, No. 12, pp. 238-250, 2018.
- [8] J. S. Kim, S. K. Han, S. W. Kim, S. W. Jung, J. M. Yang, W. D. Jang, J. N. Kim, M. Y. Ryu, J. T. Lee, S. J. Jun, and S. H. Kim, 2015 Education policy network training on-site support research: Development of SW education teaching and learning model, Commissioned research CR 2015-35, 2016.
- [9] M. S. Lee, “A Study on Creative and Convergent SW Education Programs for improving Computational Thinking,” *Journal of the Korea Society of Computer and Information*, Vol. 22, No. 8, pp. 93-100, 2017.
- [10] 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.
- [11] S. J. Lee, M. J. Lee, and Y. S. Park, “A Study on a Educational Model for Computational Thinking Development with Excel Program”, *Journal of Digital Contents Society*, Vol. 20, No 1, pp. 65-74, 2019.
- [12] S. H. Kim, “Analysis of Non-Computer Majors’ Difficulties in Computational Thinking Education,” *The Journal of Korean Association of Computer Education*, Vol. 18, No. 3, pp. 49-57, 2015.
- [13] M. J. Lee, “Exploring the Effect of SW Programming Curriculum and Content Development Model for Non-majors College Students: focusing on Visual Representation of SW Solutions,” *The Journal of Digital Contents Society*, Vol. 18, No. 7, pp. 1313-1321, 2017.
- [14] K. M. Kim, “A study on information literacy education for enhancing computational thinking,” *The Journal of Korean Association of Computer Education*, Vol. 20, No. 4, pp. 59-66, 2017.
- [15] K. S. Oh and S. J. Ahn, “A study on development of educational contents about computational thinking”, *The Journal of Korean Association of Computer Education*, Vol. 19, No. 2, pp. 11-20, 2016.
- [16] J. H. Park, S. H. Park, and Y. G. Seo, “Software Education and Business Model for Enforcing the Computational Thinking Ability,” *Journal of Digital Contents Society*, Vol. 19, No. 12, pp. 2297-2304, 2018.
- [17] H. J. Ju, H. M. Lee, and S. K. Choi, “Development of Coding Education to Enhance 4Cs Competency - Focusing on Liberal Education of S Junior College -,” *Culture and Convergence*, Vol. 41, No. 6, pp. 817-846, 2019.
- [18] E. S. Kang, “Structural Software Education Model for Non-majors - Focused on Python,” *Journal of Digital Contents Society*, Vol. 20, No. 12, pp. 2423-2432, 2019.
- [19] 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.
- [20] H. S. Kim and D. K. Tak, “Analysis on the Effectiveness of Life-based Coding Education Using Big Data Analysis,” *Journal of Digital Contents Society*, Vol. 20, No. 10, pp. 1943-1952, 2019.
- [21] K. M. Kim and H. S. Kim, “A Case Study on Necessity of Computer Programming for Interdisciplinary Education,” *Journal of Digital Convergence*, Vol. 12, No. 11, pp. 339-348, 2014.
- [22] Y. S. Lee, “Python-based Software Education Model for Non-Computer Majors,” *Journal of the Korea Convergence Society*, Vol. 9, No. 3, pp. 73-78, 2018.
- [23] J. Y. Seo, “A Case Study on the Teaching and Learning Method of SW Education for Data Analysis Problem Solving,” *Journal of Digital Contents Society*, Vol. 20, No. 10, pp. 1953-1960, 2019.
- [24] S. B. Lee and S. S. Koh, “The Effects of the Mathematical Program, DM Based on Coding Instruction Using Python,” *Journal of Educational Research in Mathematics*, Vol. 28, No. 4, pp. 479-499, 2018.
- [25] K. E. Park and S. G. Lee, “Teaching and Learning of University Calculus with Python-based Coding Education,” *Journal Korea Soc. Math. Ed. Ser. E: Communications of Mathematical Education*, Vol. 33, No. 3, pp. 163-180, 2019.
- [26] Kwangil Ko, “A Study on the EPL using Instructional Model of SW Major’s Programming Class,” *The Journal of Digital Contents Society*, Vol. 19, No. 5, pp. 891-898, 2018.



강환수(Kang, Hwan Soo)

1991년 : 서울대학교 대학원 (이학석사)

2002년 : 서울대학교 대학원 (공학박사수료-컴퓨터그래픽스)

1992년~1998년: 삼성에스디에스

1998년~현 재: 동양미래대학교 컴퓨터정보공학과 교수

※관심분야 : 컴퓨터교육, 객체지향, 프로그래밍언어



이종만(Lee, Jong Man)

1997년 : KAIST (공학석사)

2007년 : 서강대학교 대학원 (경영학박사-경영정보시스템)

1993년~2001년: LG-EDS시스템

2001년~현 재: 동양미래대학교 경영정보학과 교수

※관심분야 : 정보기술 관리 및 활용



김희천(Kim, Hee Chern)

1989년 : 서울대학교 (이학사)

1991년 : 서울대학교 대학원 (이학석사)

1998년 : 서울대학교 대학원 (이학박사-소프트웨어 공학)

2004년~현 재: 한국방송통신대학교 컴퓨터과학과 교수

※관심분야 : 머신러닝, 소프트웨어 공학, 컴퓨터교육