

비전공자 프로그래밍 교육에서 학습 목표가 교육 만족도와 학습성과에 미치는 영향 분석

구 은 희¹ · 서 주 영^{2*}¹아주대학교 다산학부대학 조교수^{2*}아주대학교 다산학부대학 부교수

Analyzing the Impact of Learning Objectives on Education Satisfaction and Learning Outcomes in Non-Major Programming Education

Eun-Hee Goo¹ · Joo-Young Seo^{2*}¹Assistant Professor, Da-San University College, Ajou University, Suwon 16499, Korea^{2*}Associate Professor, Da-San University College, Ajou University, Suwon 16499, Korea

[요 약]

본 연구는 대학의 소프트웨어 비전공자를 대상으로 하는 프로그래밍 기초교육에서의 학습 목표의 중요성에 주목하고 있다. 파이썬 프로그래밍 역량 배양의 일반적 학습 목표를 갖는 ‘대화형 프로그래밍’ 교과목과 파이썬 프로그래밍을 학습하지만, 지능형 프로그래밍 활용을 위한 구체적 학습 목표를 갖는 ‘인공지능의 이해 및 활용’의 두 교과목을 대상으로, 학습 목표의 차이가 교육 만족도와 학습성과에 어떤 영향을 미치는지 비교 분석하였다. 그 결과 구체적 학습 목표로 설계된 ‘인공지능의 이해 및 활용’ 과목이 교육 만족도와 학습성과 측면 모두에서 더 우수한 교육 효과를 보임을 확인할 수 있었다. 특히 이 교과목에서는 프로그래밍 교육에선 잘 활용하지 않는 세미나 수업을 도입하여 학생 스스로 전공적 연계성과 프로그래밍의 필요성에 공감하며, 프로그래밍에 대한 학문적 호기심과 학습 동기를 높일 수 있었고, 그 결과 프로그래밍 역량도 강화된 것으로 나타났다. 본 연구는 비전공자를 대상으로 하는 프로그래밍 교육에서 학습 목표의 구체화와 전공 연계성을 강조한 학습 활동이 효과적일 수 있음을 보여준다.

[Abstract]

This study focuses on the importance of learning objectives in introductory programming education for non-major students in university. Two courses were examined: a general programming course with the common learning objective of cultivating Python programming skills and a course titled “Understanding and Application of Artificial Intelligence,” which has specific learning objectives for utilizing intelligent programming. The study compared how differences in learning objectives affect educational satisfaction and learning outcomes between these two courses. The results confirmed that the course “Understanding and Application of Artificial Intelligence” demonstrated superior educational effects in terms of both satisfaction and learning outcomes. Particularly, this course introduced seminar-style classes, which are not commonly utilized in programming education. This approach allowed students to empathize with the relevance of programming for their major and enhance academic curiosity and motivation for learning programming, ultimately resulting in improved programming skills. This research highlights that in programming education for non-majors, emphasizing the specificity of learning objectives and their relevance to the major can be effective in enhancing learning activities.

색인어 : 학습 목표, 대학 기초교육, 교육 효과성, 소프트웨어 비전공자, 프로그래밍 교육**Keyword** : Learning Objectives, University Foundation Education, Educational Effectiveness, Software Non-Majors, Programming Education<http://dx.doi.org/10.9728/dcs.2024.25.4.1069>

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 21 February 2024; Revised 18 March 2024

Accepted 25 March 2024

***Corresponding Author, Joo-Young Seo**

Tel: +82-31-219-3364

E-mail: jyseo@ajou.ac.kr

1. 서론

최근 급속도로 발전하는 인공지능 시대를 맞이하여 소프트웨어 교양 교육에 대한 필요성은 크게 공감하고 있으며, 초등학교부터 대학교에 이르기까지 소프트웨어 기초교육에 대한 논의가 지속되고 있다[1],[2]. 대학의 경우, 2015년 과학기술정보통신부 주관인 SW중심대학 사업을 시작점으로 전공과 상관없이 전교생이 필수로 소프트웨어 교육을 의무수강해야 하는 ‘비전공자 소프트웨어 기초교육’이 시행되어왔다. 사업 초기엔 SW중심대학으로 선정된 대학 대부분이 ‘컴퓨팅사고’ 또는 ‘프로그래밍’ 교과목을 전교생이 동일 방식으로 수강하는 소프트웨어 기초교육이 일반적이었다[3].

초기엔 ‘컴퓨팅사고’와 관련해선 대학생이기에 초중고 학생보다는 다소 복잡한 알고리즘을 교육하여야 한다는 인식 때문에 학습에 어려움을 준다든지, ‘프로그래밍’에 대해선 전공자보다는 쉬운 난이도의 교육을 위해 스크래치나 엔트리와 같은 블록 코딩 언어를 교육하여 오히려 학습 의욕을 떨어뜨리는 등 소프트웨어 기초교육의 중요성은 공감하지만, 학습 효과를 살릴 수 있는 교육 콘텐츠 구성엔 시행착오가 많았다. 또한 전교생이 동일한 교육을 받는 방식은 관심사와 기초수준이 다른 소프트웨어 비전공자들의 학습 동기를 불러일으키기 힘들기에 학습의 어려움을 더욱 배가시켰다[4]-[8].

최근 이러한 문제점들은 자신의 전공과 관련성 높은 소프트웨어 역량을 교육하는 ‘전공맞춤형 소프트웨어 기초교육’으로 학습 동기를 높임으로 해소되고 있다. 전공맞춤형 소프트웨어 기초교육은 전교생이 동일 교과목을 학습했던 예전과는 다르게 계열별 또는 학과별로 유용한 소프트웨어 역량으로 교과과정을 설계하는 것을 의미하며 동일 교과목을 학습하는 경우보다 학습 동기와 학습 효과가 모두 더 높다고 보고되고 있다[8]. 전공계열은 인문계열, 이공계열의 일반적 양분화에서 ‘경상계열, 인문계열, 사회과학계열, 자연과학계열, 공학계열, 의학계열, 예체능계열’ 등으로보다 세분화되고 있고, 교육 콘텐츠 역시 ‘컴퓨팅사고, 데이터분석, 프로그래밍, 인공지능, 소프트웨어 리터러시, 컴퓨터개론’ 등으로 다양화되는 추세이다[9]-[16].

전공맞춤형 소프트웨어 기초교육이 학생들의 학습 동기와 자신감을 높이고 교육 효과도 긍정적으로 변화시켰다는 연구[16]에 따르면, 교육 초기엔 인문계열이 이공계열보다 소프트웨어 역량이 상대적으로 낮으며 전공계열별로도 소프트웨어 역량에 차이를 보였으나, 교육 후 역량의 상승과 함께 전공계열 간의 유의미한 차이가 사라진다고 보고하고 있다. 특히 이러한 교육 효과성 변화는 경상계열 > 사회과학계열 > 인문계열의 순으로 유의미한 상승을 보였고, 교육 콘텐츠로는 데이터분석 > 컴퓨팅사고가 유의미하게 상승했다고 한다.

그러나, 이공계열 학생들의 주 교육 콘텐츠인 프로그래밍과 관련해선 교육 전, 후의 학생들의 소프트웨어 역량에 큰 변화가 없었으며, 교육 후의 교육 효과 역시 크지 않은 것으

로 보고되었는데, 이는 현재 비전공자 소프트웨어 기초교육에 대한 논의가 상대적으로 인문계열 학생들에게 집중되고 있는 점, 이공계열은 오래전부터 프로그래밍 교육이 필수로 운영된 경우가 많기에 교육 콘텐츠 면에서 큰 고민이나 변화가 없었던 것이 이유일 것으로 생각된다.

그러나 교육 후에 별다른 역량 변화가 없다는 것은 큰 문제를 의미하기에 본 논문에서는 비전공자를 대상으로 하는 프로그래밍 교육에 있어 교육 효과를 높이는 방안으로 ‘학습 목표’의 중요성에 대해 논의하려 한다.

일반적으로 프로그래밍 교육은 프로그래밍 언어의 문법을 정확히 이해하고 이를 통해 프로그래밍 문제를 효율적으로 해결하는 것이 주요 학습 목표이다. 그러나 이러한 학습 목표는 실제 한 학기 1개 과목 수강으로 달성되는 목표는 아니며, 연결성 있는 교과과정의 큰 목표(프로그래밍 문제를 효율적으로 해결)를 위한 기초 단계에 해당하는 부분 목표(프로그래밍 언어의 문법을 정확히 이해)로 단계적으로 성취될 수 있다.

그러나 비전공자의 소프트웨어 기초교육의 경우 1과목 또는 2과목 정도를 수강하는 것이 일반적이기에 전공자와 같이 연결되는 교과과정을 기대하기 어렵다. 그렇기에 기존의 프로그래밍 교육의 학습 목표를 그대로 따르면 소프트웨어 비전공 학생 입장에서 부분 목표인 프로그래밍 문법을 이해하는 것에 그치고, 주 목표인 프로그래밍 문제 해결은 제대로 학습하지 못하기에 궁극적으로 무엇을 위해 프로그래밍을 배우는 것인지 방향성을 잃게 되고 학습 동기나 성취도 면에서도 학습 효과가 떨어지는 것은 당연할 것이다.

본 논문에서는 A 대학의 이공계열을 위한 파이썬 프로그래밍 교과목 사례를 통해 학습 목표 차이에 따른 비전공자 프로그래밍 교육 효과를 비교 분석한다. 비교 대상 교과목은 ‘대화형 프로그래밍’과 ‘인공지능의 이해 및 활용’으로 두 교과목 모두 파이썬 프로그래밍이 주요 교육 내용이나, ‘대화형 프로그래밍’은 파이썬 언어를 활용하여 프로그래밍 문제 해결 역량을 키우는 것이 학습 목표이며 일반적인 프로그래밍 교육이 그래왔듯이 문법을 중심으로 한 교과과정으로 진행된다. ‘인공지능의 이해 및 활용’ 교과목은 인공지능 프로그래밍을 목표로 파이썬 언어를 학습하는 것으로 인공지능이란 구체적인 문제 도메인을 목표로 정의한 것이 주요 차이이다.

우리는 학습 목표에 차이를 보이는 이 두 교과목의 강의평가 결과와 학생 성적을 비교 분석함으로써 학습 목표의 차이에 따른 프로그래밍 교육의 만족도와 학습성적을 분석한다.

본 논문은 다음과 같이 구성한다. 2장은 관련 연구로 대학의 소프트웨어 비전공자 프로그래밍 기초교육의 현황과 함께 A 대학의 현황을 구체적으로 살펴본다. 3장에선 A 대학의 비전공자 프로그래밍 교육에 있어 학습 목표가 다른 2개 교과목을 비교 분석함으로써 학습 목표에 따라 교육 만족도와 학습성과의 차이를 논의한다. 끝으로 4장에선 본 사례 연구 결과를 토대로 한 비전공자 프로그래밍 교육에 대한 제언으로 맺는다.

II. 소프트웨어 비전공자 프로그래밍 기초교육 현황

2-1 SW 중심대학의 프로그래밍 기초 교육 현황

비전공자 대상의 소프트웨어 기초교육은 2015년 SW중심대학을 주축으로 시작되어 현재 대부분의 대학에서 전공과 상관없이 소프트웨어 교육을 필수로 교육하고 있다. 다양한 소프트웨어 기초교육 콘텐츠 중 프로그래밍은 SW중심대학 이전부터 인기 있는 교육 콘텐츠였으며, 다음 표 1은 주요 SW중심대학의 소프트웨어 비전공자 대상 프로그래밍 기초교육 현황이다[3],[17]-[22].

표 1. 주요 SW중심대학의 비전공자 프로그래밍 기초 교육 현황
Table 1. I status of programming basic education in national program of excellence in SW

Univ.	Course	Credits	PL	Major
A	- Interactive Programming - Scientific Computing Programming - Programming Fundamentals - Understanding and Utilizing Artificial Intelligence	3	C/C++ Python Matlab	SE
B	- Problem Solving and Algorithms - Computer Programming for Engineering - Programming Fundamentals and Practical Exercises - Fundamentals and Application of Artificial Intelligence	2-3	C/C++ Python	SE
C	- Fundamentals of Web Applications - Python Programming - Computer Programming - Java Programming	3	C/C++ Python Java	SE
D	- Coding Basics in Humanities and Social Sciences - Coding Basics in Natural Sciences	3	Python	HS, SE
E	- Python Programming - Fundamentals of C Programming - Basics of Web Programming	3	Python C Java	SE
F	- Programming Fundamentals	3	Python	SE

PL: Programming Language, SE: Science and Engineering Departments, HS: Humanities and Social Sciences Departments

표 1에서 보듯이 많은 대학이 소프트웨어 비전공자를 위해 다양한 프로그래밍 교과목을 개설하고는 있으나 전공별로 1과목을 선택하여 수강하기에 프로그래밍 역량을 위해선 3학점 정도를 할애하는 것이 일반적이다.

또한 프로그래밍 교과목은 인문계열보다는 이공계열을 위해 개설되고 있으며, SW중심대학 초기에 컴퓨팅 사고력 교육

을 위한 스크래치나 엔트리와 같은 블록형 프로그래밍 언어는 배제되고 비전공자를 위해서도 C, 파이썬, 자바와 같은 범용 프로그래밍 언어가 일반적이며, 현재 이들 중 파이썬 프로그래밍 언어의 선호도가 높다. 또한 최근엔 인공지능(AI; artificial intelligence) 리터러시에 관한 관심이 높아지면서, 근래 SW중심대학으로 선정된 학교의 경우 비전공자를 위해서도 인공지능과 파이썬 프로그래밍을 함께 설계하는 교과목도 늘어나는 추세이다[3].

이렇듯 소프트웨어 비전공자 대상 프로그래밍 기초교육이 보다 다양성을 보이고 있지만, 프로그래밍 언어 종류에 따른 차이가 대부분이며, 이민정[23]의 연구에 따르면 비전공자 대상 프로그래밍 기초교육은 여전히 프로그래밍 언어의 문법과 프로그래밍 도구 학습에 시간 대부분을 투자하고 있다고 분석하고 있다. 또한 소프트웨어 전공자와 다르게 비전공 학생들은 해당 과목을 이수한 후 후속 학습을 이어가기 쉽지 않기에 이들을 위한 프로그래밍 기초교육은 프로그래밍 언어 습득에 집중하기보다는 컴퓨팅 환경에서의 문제해결력 함양을 돕는 방식으로 더 노력을 기울여야 함을 주장하고 있다.

김재경[24]의 비전공자 대상 프로그래밍 입문 교육의 학습자 어려움의 분석 결과에 따르면, 프로그래밍에 낮은 학습 동기를 가진 학습자의 경우 진도가 진행됨에 따라 학습성과는 점점 낮아지는 경향이 있으며, 적극적인 동기를 가진 집단은 전체 과정 동안 어려움을 덜 겪으며, 심리적 요소가 학습에 미치는 영향이 큼을 보고했다. 그렇기에 학생들이 학습에 흥미와 적극적 동기를 유발할 수 있도록 학습 목표를 명확히 제시하고 관심 문제로 교과 내용을 구성하는 등의 교과목 설계가 필요함을 언급하고 있다.

비전공자 대상의 프로그래밍 기초교육은 소프트웨어 전공자와는 다르게 여러 교과목으로 프로그래밍에 대한 경험을 점진적으로 쌓기 어렵기에 기존의 프로그래밍 문법을 이해하는 일반적 학습 목표가 아닌 학습자의 동기를 이끄는, 보다 구체적인 학습 목표와 운영 방안을 고민해 볼 필요가 있을 것이다.

2-2 A 대학의 프로그래밍 기초교육 현황

A 대학은 지난 2015년부터 현재까지 SW중심대학으로서 8년의 기간 동안 비전공자 대상의 소프트웨어 기초교육을 시행하고 있다. A 대학은 인문계열 학생들을 위해선 ‘컴퓨팅사고, 데이터분석’의 콘텐츠를 다루는 소프트웨어 기초 교과목을 개설하고 있으며, 특히 이공계열을 위해선 SW중심대학 선정 이전부터 C/C++, Python, Matlab과 같은 다양한 언어의 프로그래밍 교육을 운영해왔다.

A 대학의 비전공자 소프트웨어 기초교육의 교육 효과를 분석한 연구[16]에 따르면, 컴퓨팅사고나 데이터분석을 기초 교과목으로 학습한 인문, 사회과학, 경상 계열의 경우, 소프트웨어에 대한 자신감, 흥미의 소프트웨어 인식과 태도, 컴퓨팅 사고력, 소프트웨어 문해력과 같은 소프트웨어 역량이 교육 이후 모두 높게 상승하는 학습 효과를 보인 반면, 프로그래밍

교과목을 학습한 이공계열의 경우 교육 후 소프트웨어 역량이 평균적으로 상승하기는 했으나 유의미한 차이를 보이지 않았다고 보고했다. 인문, 사회과학, 경상 계열의 높은 교육 효과는 컴퓨팅사고나 데이터분석 교과목이 여러 시행착오를 거쳐 이들 계열 학생의 학습 동기를 자극할 관심 문제와 사례를 기반으로 교과과정이 개선을 거듭한 성과였으며, 반면 이공계열의 프로그래밍 교과목의 경우, 기존의 프로그래밍 문법 위주의 교과과정을 크게 벗어나지 못한 결과로 분석하였다.

본 논문은 소프트웨어 비전공자의 프로그래밍 기초교육 효과를 높이는 방법의 하나로 학습 동기를 자극하기 위해 학습 목표의 구체적 제시와 교과 운영 방안을 제안하며, 이에 따른 교육 효과를 분석한다.

III. 학습목표에 따른 프로그래밍 교육효과 비교 연구

본 사례 연구는 A 대학의 이공계열 신입생들을 위해 개설되는 파이썬 프로그래밍 교육 사례로 소프트웨어 비전공자를 위한 프로그래밍 기초교육에 있어서 학습 목표가 다른 경우 교육 만족도와 학습성과에도 차이를 보이는지를 비교 분석한 연구이다.

비교 대상 교과목은 ‘대화형 프로그래밍’과 ‘인공지능의 이해 및 활용’으로 두 교과목 모두 핵심 교육 콘텐츠는 공통적으로 파이썬 프로그래밍 활용 능력을 키우기 위한 내용으로 구성된다. 하지만 ‘대화형 프로그래밍’ 교과목은 소프트웨어로 문제 해결을 위한 방안으로 파이썬 프로그래밍 역량을 키우는 것이 주 학습 목표이며, 대학의 프로그래밍 수업에 일반적으로 적용되는 커리큘럼으로 설계된 교과이다. ‘인공지능의 이해 및 활용’은 인공지능에 대한 이해와 이를 위한 지능형 프로그램 구현을 위해 파이썬 프로그래밍을 학습하는 것으로 학습 목표에 차이를 보인다. 즉 파이썬을 사용하여 실제 인공지능 기술을 활용하는 방법을 학습하며, 프로그래밍 문법보다는 무엇을 개발해야 하는지, 어떻게 사용되었는지 등에 대한 것들을 학습하는 교과이다.

본 연구에선 학습 목표가 지향하는 바가 다른 이 두 교과목에 대해 학생들의 강의평가 결과를 토대로 교육에 대한 만족도를 비교하며, 성적 분석을 통해 파이썬 프로그래밍 역량에 대한 학습성과에 차이가 있는지를 비교 분석한다.

3-1 연구 대상

본 연구는 동일 학기에 동일 교수자에 의해 운영된 ‘대화형 프로그래밍’ 수업을 수강한 자연대학 학생 46명과 ‘인공지능의 이해 및 활용’ 수업을 수강한 공과대학 학생 33명을 연구 대상으로 한다. 교육 만족도 분석을 위해선 강의평가를 담당하는 부서를 통해 학생들이 강의의 만족도를 평가한 익명의 자료를 제공받아 분석하였다. 학습성과의 경우, 전체 성적을 대상으로 분석한 것이 아닌 학생들의 성적 평가 요소 중 파이

썬 프로그래밍 역량을 평가한 자료를 선별하여 분석하였다. 두 교과목의 학습 목표가 다른 만큼 평가 요소도 차이를 보이는 부분이 있지만, 두 교과목 모두 파이썬 프로그래밍 역량 배양이란 기본 학습 목표는 공통인 만큼 교과과정 중 파이썬 프로그래밍과 관련해선 동일한 내용으로 학습했으며, 시험 역시 두 교과목 모두 동일 기준으로 평가되었다.

1) 대화형 프로그래밍 교과목

‘대화형 프로그래밍’ 교과목은 소프트웨어 비전공 학생들을 대상으로 소프트웨어를 활용한 문제 해결을 위해 필요한 기본 지식과 함께 파이썬 언어를 이용하여 자신만의 소규모 프로그램을 개발하고 이를 활용할 수 있는 기회를 제공하는 것을 학습 목표로 하는 과목이다. 이를 위해 학생들이 학습하는 내용은 표 2에서 보듯이 3주차까지 수업 진행에 필요한 프로그래밍 환경, 소프트웨어, 컴퓨팅사고 및 이를 활용한 문제 해결 방법에 대한 기초지식을 습득한다. 4주부터 7주차까지의 4주의 기간 동안 파이썬 언어의 기초 문법을 익히고 이후 3주간인 10주차까지는 파이썬 문법의 활용을 배우게 된다. 11주차부터는 파이썬이 제공하는 유용한 자료구조를 이용하여 소규모 프로그램을 개발하여 응용할 수 있는 방법을 배우

표 2. 대화형 프로그래밍 교과과정

Table 2. Interactive programming course curriculum

Week	Topic	Contents
1	Course Overview / Software Overview	- Explanation of the Course - Computer & Soft Ware
2	Software Overview	- Soft Ware - Programming as a tool for understanding Software
3	Software Overview	- Computing Thinking - Problem Solving
4	Python Basics	- Python Overview - Using Python - Python Variable
5	Python Basics	- Python Data Type - Python Operator
6	Python Basics	- Python Input - Python Output
7	Python Basics	- Python Modules
8	Python Applications	- Python Selection Statements
9	Python Applications	- Python Loop Statements
10	Python Applications	- Python Functions
11	Python Data Structures	- String - Tuple
12	Python Data Structures	- List
13	Python Data Structures	- Dictionary
14	Python Data Structures	- Set



*The assignment is presented in its original form as written by the students.

그림 1. ‘대화형 프로그래밍’ 과목의 수업 요약과제
Fig. 1. Summary assignment for ‘interactive programming’ course

게 된다. 다음 표 2는 ‘대화형 프로그래밍’ 수업의 중간/기말 시험기간을 제외한 14주 수업 커리큘럼을 나타낸 것이다.

이 교과목의 수업 운영은 파이썬 문법의 소개와 문법에 해당하는 사용법을 알려주고 이를 실습할 수 있는 예제 문제와 활용 문제를 푸는 일반적인 프로그래밍 수업 방식을 기본적으로 따르고 있다. 수업 운영의 특징은 학생들의 수업 이해도를 확인하기 위하여 수업의 마무리로 이해한 내용을 요약하는 수업 요약과제를 제출하는 것이다. 이 과제는 해당 수업에서 배운 이론적인 내용과 실습 내용을 바탕으로 정리하되, 학생 스스로가 이해한 만큼, 프로그래밍 실습 또한 할 수 있는 만큼의 내용을 정리하여 제출하는 과제이다. 그림 1은 ‘대화형 프로그래밍’ 과목에서의 학생들이 제출한 수업 요약과제의 예로 자료형, 입출력, 반복문, 문자열 파트를 정리하여 제출한 것이다.

2) 인공지능의 이해 및 활용 교과과정

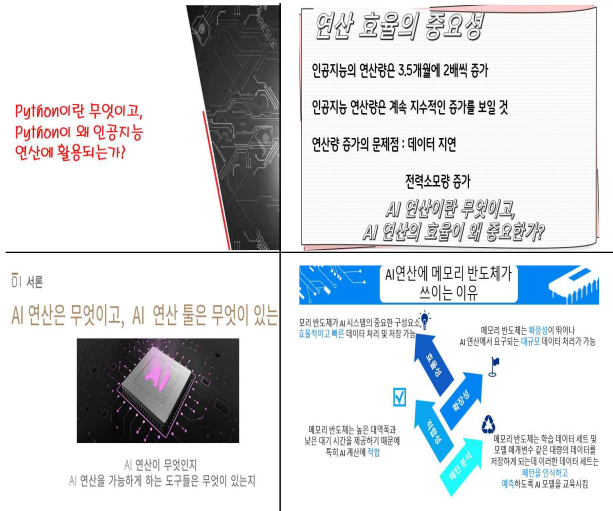
‘인공지능의 이해 및 활용’ 교과목은 인공지능을 이해하고, 자신의 전공 분야에 활용할 수 있는 지능형 프로그래밍 역량을 갖추는 것을 학습 목표로 하는 과목이다. 이를 위해 학생들은 인공지능의 기본 개념을 이해하는 것을 시작으로, 다음 표 3과 같이 5주차까지 기초 인공지능 기술과 핵심 알고리즘, 응용 방법, 인공지능 서비스 등의 소개를 통하여 인공지능에 관한 이

론적 핵심을 이해하고, 인공지능 기법을 이해하기 위한 시연을 통하여 프로그래밍에 대한 필요성을 학습한다. 6주차부터 11주차까지는 파이썬 언어의 기초 문법부터 자료구조에 이르기까지 파이썬 프로그래밍 역량 배양을 위한 시간으로 대화형 프로그래밍 교과목의 4주차부터 11주차까지의 내용과 동일하게 학습한다. 이후 마지막 14주차까지는 파이썬을 이용한 인공지능의 실제 활용을 학습함으로써 단순히 파이썬이란 프로그래밍 언어를 아는 것을 넘어 현실 세계와 자신의 전공에서의 활용까지 이해할 수 있도록 설계한 것이 특징이다.

이 교과목의 수업 운영에 있어 특징적인 것은 파이썬 학습의 중간 10주차에 인공지능 기술 및 파이썬이 학생들의 전공에 어떻게 이용되는지, 어떠한 영향을 미치는지에 대한 세미나를 진행하는 시간이다. 이 세미나 시간은 프로그래밍을 학습하다 보면 어느 순간 무엇을 위해 프로그래밍을 배우는지에 관한 학습 목표가 모호해지는 순간이 생기는데, 이를 보완하고 학생 스스로 학습 동기를 찾아보는 시간이라 할 수 있다.

표 3. 인공지능의 이해 및 활용 교과과정
Table 3. Understanding and utilizing artificial intelligence course curriculum

Week	Topic	Contents
1	Course Overview / AI Overview	- Explanation of the Course - History of AI - Python as a tool for understanding AI
2	AI Techniques / Experience AI Service	- AI's core concepts and technologies - Experiencing webs or apps utilizing AI technology
3	Knowledge and Rule-based AI	- The core concept of Knowledge and Rule-based AI
4	AI Learning method	- The definition of supervised, unsupervised, and reinforcement learning
5	Deep Learning / AI Ethics	- The definition of machine learning - AI Ethics - AI and future
6	Python Basics	- Examples of AI Projects
7	Python Basics	- Python Overview - Using Python
8	Python Basics	- Variables, Input, Output, Operator
9	Python Application	- Flow Control; Selection, Loop
10	AI technique Seminar	- AI techniques for the major
11	Python Data Structures	- String, Tuple, List, Dictionary, Set
12	Python for AI	- Python Library for AI - Numpy, Pandas, Matplotlib Practice
13	Python for AI	- AI and data
14	Python for AI	- Artificial Neural Network



*The assignment is presented in its original form as written by the students.

그림 2. 세미나 수업의 발표 자료

Fig. 2. Seminar Class Presentation Materials The assignment is presented in its original form as written by the students.

프로그래밍에 대한 지식이 미흡한 비전공 학생들이 프로그래밍 언어의 문법 자체에만 몰입해서 성취해야 할 학습 목표를 놓치는 것을 막기 위해 프로그래밍 학습의 중간에 인공지능과 파이썬의 관계라든지, 자신의 전공 분야에서 인공지능 프로그래밍의 역할이나 최신 동향과 같은 주제로 세미나를 진행함으로써 프로그래밍 학습에 좀 더 동기 부여를 할 수 있을 것으로 기대하였다. 그림 2는 세미나 수업에서 학생들의 발표에 사용된 발표 자료의 예로 ‘Python과 인공지능을 포함하여 전공과 관련된 이슈’를 조사하여 만든 발표 자료이다.

다음 표 4는 ‘대화형 프로그래밍’과 ‘인공지능의 이해 및 활용’ 교과목의 주요 차이점인 ‘학습 목표, 수업에서 학습하는 내용, 수업 운영 방법과 주요 학습 활동 등 주요 차이점을 요약한 표이다.

표 4. 대화형 프로그래밍과 인공지능의 이해 및 활용 교과목 비교
Table 4. Comparative analysis of IP and AI courses

Contents	IP	AI
Learning Objectives	Developing Programming Competence for Problem Solving with Computational Thinking	Fostering Intelligent Programming Competence for Artificial Intelligence
Learning Content	Software Overview Python Programming	AI Overview Python Programming
Class Management	Syntax – Practical Exercises – Class Summaries	AI Techniques – Demonstrations – Syntax – Practical Exercises
Learning Activities	Class Summary Assignment	Seminar Assignment

IP: Interactive Programming,
AI: Understanding and Utilizing Artificial Intelligence

3-2 연구 문제

본 사례 연구는 두 교과목의 교육 효과를 살피기 위해 강의 평가를 기반으로 한 ‘교육 만족도’와 성적을 토대로 한 ‘학습 성과’를 연구 문제로 차이를 비교 분석한다.

1) 연구 문제 1: 교육 만족도 분석

교육 만족도는 다음 표 5와 같은 강의평가 설문을 토대로 ‘수업 준비(C1), 수업 진행(C2), 강의 구성(C3), 교수 방법(C4), 과제 및 평가(C5), 강의 운영(C6)’을 묻는 6개의 객관식 문항으로 분석한다. 각 문항은 1점 ‘매우그렇지않다’, 2점 ‘그렇지않다’, 3점 ‘보통이다’, 4점 ‘그렇다’, 5점 ‘매우그렇다’의 5점 리커트 척도로 측정하였다.

표 5. 강의평가 설문

Table 5. Lecture evaluation questionnaire

No	Contents
C1	The lesson plan is designed to enhance understanding of the course's nature and assist in preparing for the class, taking into consideration the subject's characteristics.
C2	The lecture was conducted to cultivate academic interest and curiosity.
C3	The course content was systematically organized and presented.
C4	The instructor's teaching methods(lecture, discussion, demonstration, etc.) were appropriate for this course.
C5	The assignments effectively complemented the course content, enhancing the overall learning experience.
C6	The lectures were appropriate and proved beneficial for learning.

2) 연구 문제 2: 학습성과 분석

학습성과는 파이썬 프로그래밍 역량 평가와 관련된 성적으로 분석한다.

두 교과목은 학습 목표에 차이가 있는 만큼 전체 평가 요소 구성에도 차이를 보이는데 ‘대화형 프로그래밍’의 경우, ‘출석, 수업 활동(수업 요약과제 20회; A4 1장), 필기시험 2회(중간고사 및 기말고사)’로 평가한다. ‘인공지능 이해 및 활용’ 교과목은 ‘출석, 세미나 과제 1회(보고서 및 발표), 필기시험 2회(중간고사 및 기말고사)’로 평가한다.

두 교과목의 평가 요소 면에서 큰 차이점은 ‘대화형 프로그래밍’ 교과목에선 수업 활동으로 수업 시간 중 수업 내용을 잘 이해했는지를 확인하는 수업 요약과제가 학습성과를 높이기 위한 주요 평가 요소로 설계되었고, ‘인공지능의 이해 및 활용’ 교과목은 파이썬과 인공지능이 자신의 전공 분야에 어떻게 활용되고 있는지를 조사하여 발표함으로써 스스로 학습 동기를 찾아보는 세미나 과제를 학습 활동으로 포함하고 있다는 점이다.

본 연구에선 학습성과를 높이기 위한 두 교과목의 학습 활동의 차이가 실제 파이썬 프로그래밍 역량에는 어떻게 반영

되어 나타나는 지를 확인해 볼 것이다. 이때 평가 요소의 다른 부분의 성적은 배제하고, 파이썬 프로그래밍 역량을 평가했던 필기시험 성적만을 대상으로 학습성취를 비교 분석하며, 이들 분석에 사용된 성적 자료는 두 교과목 모두 동일 시험으로 평가되었음을 전제한다.

3-3 연구 결과

본 연구에서 비교하는 두 교과목의 공통적인 학습 내용은 프로그래밍 활용 능력의 배양을 위해 파이썬 언어를 기초 문법에서부터 시작하여 활용하는 방법을 배우는 것이다. 그러나 동일한 내용을 학습하더라도 두 과목의 주요 학습 목표는 차이가 있는데, ‘대화형 프로그래밍’ 교과목에서는 소프트웨어를 활용한 문제 해결을 위한 방안의 하나로 파이썬 언어를 학습하는 자체가 주요 학습 목표이고, ‘인공지능의 이해 및 활용’ 교과목에서는 인공지능을 위한 지능형 프로그램을 개발하기 위해 파이썬 언어를 학습하는 것으로 학습 목표에 차이가 있다. 또한 비교 대상 두 교과목의 차이는 ‘대화형 프로그래밍’은 매 수업마다 학습 이해도를 확인하는 수업 요약과제를 주요 수업 활동으로 운영하였고, ‘인공지능의 이해 및 활용’은 자신의 전공에서 프로그래밍 필요성 및 활용 방법 등에 대한 의견을 나누는 세미나 수업 시간을 두어 프로그래밍 언어를 익히는 과정 중에 다시 한번 학생 스스로 프로그래밍 역량의 중요성을 깨닫고 학습 동기를 높일 수 있도록 유도한 부분이다. 이러한 차이를 갖는 두 교과목을 앞서 정의한 연구 문제에 따라 비교 분석한 결과는 다음과 같다:

1) 연구 문제 1: 교육 만족도 분석

‘대화형 프로그래밍’ 교과목(IP; interactive programming) 과 ‘인공지능의 이해 및 활용’ 교과목(AI)의 교육 만족도 차이를 독립표본 t-검정으로 분석하였다. 다음 표 6의 분석 결과를 보면, 모든 문항에서 ‘인공지능의 이해 및 활용’이 ‘대화형 프로그래밍’ 교과목보다 높은 평균을 보였으며, 수업 준비(C1) 항목을 제외하고는 통계적으로 유의미한 차이(p-value<0.05)를 보인다. 평균이 유의미한 차이를 보이지 않은(p-value>0.05) 수업 준비(C1) 문항의 경우, ‘대화형 프

로그래밍’은 평균 4.51, ‘인공지능의 이해 및 활용’은 평균 4.67로 두 교과목 모두 높은 점수를 보이고 있으며, 수업 계획이 과목의 특성을 고려하여 과목의 성격에 대한 이해를 높이도록 잘 설계되어 있으며, 학생들의 수업 준비에 충분히 도움이 되었고 두 교과목 모두 비슷한 수준의 만족도를 보인다는 것을 의미한다.

나머지 항목들(C2-C6)은 ‘인공지능의 이해 및 활용’이 ‘대화형 프로그래밍’보다 평균이 통계적으로도 유의미하게 높게 나타났다(p-value<0.05). 유의미한 차이를 보인 항목 중 가장 높은 평균을 보인 항목은 ‘강의 내용이 체계적으로 구성되어 제시되었다’의 강의 구성(C3) 항목과 ‘강의는 적절했고 학습에 도움이 되었다’의 강의 운영(C6) 항목으로 ‘인공지능의 이해 및 활용’의 경우 두 항목 모두 평균 4.73으로 매우 높은 만족도를 보인다.

강의 구성(C3) 항목의 경우 ‘대화형 프로그래밍’ 교과목 역시 평균이 4.38로 높은 평균값을 보이는데, 두 교과목의 수업 내용에서 많은 부분을 차지하는 파이썬 프로그래밍 부분의 강의 구성이 유사한 면이 있었기 때문인 듯하다. 평균 차이를 보면 ‘강의 내용이 체계적으로 구성되어 제시되었다(C3)’의 경우는 t-value가 -3.16이고, ‘강의는 적절했고 학습에 도움이 되었다(C6)’는 -7.29로 ‘대화형 프로그래밍과의 차이’는 강의 구성(C3)보다 강의 운영(C6)의 항목이 더 큰 차이를 보임을 알 수 있다. 강의 운영 측면에서의 차이는 ‘대화형 프로그래밍’ 교과목의 수업 요약과제의 학습 활동보다 ‘인공지능의 이해 및 활용’의 세미나 활동이 학생들의 전공에서 필요한 프로그래밍의 중요성을 깨닫게 하고, 이를 통해 다양한 학습 경험을 제공하여 프로그래밍과 인공지능에 대한 흥미를 유발하여 학습 목표에 긍정적인 영향을 미쳤을 것으로 생각된다.

두 번째로 ‘인공지능의 이해 및 활용’ 과목의 평균이 높은 항목은 ‘강사의 교수법(강의, 토론, 시연 등)이 본 강좌에 적합했다’의 교수 방법(C4) 항목인데, 평균 4.48, p-value<0.05로 ‘대화형 프로그래밍’과 유의미한 평균 차이를 보이며, 이 역시 프로그래밍 문법을 단계적으로 배우고 요약과제를 통해 평가하는 방식보다는 파이썬으로 구현된 인공지능 알고리즘을 시연해 본다는 것, 최신 동향을 찾아 학생들과 발표를 통해 공유하고 토론하는 등의 시간을 갖는 교수 방법이 유효했음을 보여주고 있다.

‘과제는 수업내용을 적절히 보완하여 학습효과를 높이는데 도움이 되었다’의 과제 및 평가(C5) 항목의 두 과목의 평균은 4.15와 4.55로 강의 운영(C6) 항목 다음으로 높은 차이를 보였는데 이는 ‘대화형 프로그래밍’에선 수업내용을 정리하여 제출하는 수업 요약과제가 이 교과목의 과제인 반면, ‘인공지능의 이해 및 활용’ 교과에서는 10주차에 진행되는 세미나 수업이 있어 전공에서 필요한 인공지능과 파이썬 활용을 조사하여 발표하는 과제가 진행되는 것이 만족도 점수의 차이를 보여주는 것으로 생각된다. t 값 -3.81로 통계적으로 유의미한 차이를 보이고 있음을 알 수 있다.

마지막으로 ‘강의는 학문적인 흥미와 호기심을 갖도록 진

표 6. 강의평가 분석 결과

Table 6. Results of lecture evaluation analysis

No	IP	AI	t-value	p-value
C1	4.51	4.67	-0.07	0.95e-01
C2	4.29	4.52	-4.06	1.14e-04
C3	4.38	4.73	-3.16	2.19e-03
C4	4.28	4.48	-4.45	2.89e-05
C5	4.15	4.55	-3.81	2.75e-04
C6	4.08	4.73	-7.29	2.25e-10

IP: Interactive Programming,

AI: Understanding and Utilizing Artificial Intelligence

행하였다'의 수업 진행(C2) 항목의 두 교과목의 평균값도 통계적으로 유의미한 차이(p-value<0.05)를 보이고 있다. 이와 같은 결과를 나타내는 이유는 '대화형 프로그래밍' 과목의 경우 소프트웨어를 활용한 문제 해결을 위한 방안의 하나로 파이썬 언어를 학습하는 것으로 수업 초반에는 소프트웨어와 컴퓨팅사고, 문제 해결에서의 프로그래밍과 같은 소프트웨어 기초적 이론을 배운 후 중후반부터는 파이썬 언어의 문법을 배우는 데 집중하고 있는 반면, '인공지능의 이해 및 활용' 교과목에서는 수업 전체에 걸쳐 AI 기초지식으로 시작하여 파이썬을 학습하는 중간에도 AI와 파이썬의 관계를 성찰하고 후반에는 AI를 활용하기 위하여 파이썬을 어떻게 사용하는지의 방법에 이르기까지 교과과정 전반에 걸쳐 AI와 파이썬을 함께 학습할 수 있었기 때문이라 생각된다.

2) 연구 문제 2: 학습성과 분석

다음 표 7은 두 교과목의 학습성과 차이를 분석한 결과이다. 학습성과는 파이썬 프로그래밍 역량을 평가하기 위한 중간고사와 기말고사의 시험 항목 중 두 과목에서 동일한 시험 문제로 제시한 파이썬 기초(8문제), 활용(12문제), 자료구조(10문제)의 문제들을 추출하여 항목별로 나누어 비교하였다.

표 7. 학생 성적 항목 분석 결과
Table 7. Results of student grade item analysis

Content	IP	AI	t-value	p-value
Python Basics	57	54	0.66	5.10e-01
Python Applications	49	57	-2.18	3.26e-02
Python Data Structures	62.5	70	-2.41	1.82e-02

파이썬 기초 항목에 대한 성적은 '대화형 프로그래밍(IP)' 교과목의 평균이 57점으로 '인공지능의 이해 및 활용(AI)' 교과목의 평균인 54점보다 높은 점수를 보였다. '대화형 프로그래밍'의 경우 파이썬 기초 학습에 '인공지능의 이해 및 활용'보다 2주 정도 더 할애되었고, 수업 후 요약과제를 통해 수업 시간 중 스스로 이해도를 확인하는 활동도 주어졌기에 평균이 더 높게 나온 이유가 되는 듯하다. 그러나 p-value>0.05로 이러한 평균의 차이가 통계적으로 유의하지는 않으며, 두 교과목의 파이썬 기초 역량에 대해선 유사한 성취도를 보인다고 분석된다.

좀 더 프로그래밍 난이도가 높아지는 파이썬 활용과 자료구조에선 '인공지능의 이해 및 활용' 교과목이 '대화형 프로그래밍'보다 평균이 모두 높고, 통계적으로도 유의미한 차이를 보였다. 파이썬 활용 항목에선 '인공지능의 이해 및 활용' 교과목의 평균은 57점, '대화형 프로그래밍'은 49점이며, 파이썬 자료구조 항목도 또한 '인공지능의 이해 및 활용'은 70점으로 '대화형 프로그래밍'의 62.5점보다 평균적으로 더 높은 성적과 p-value<0.05로 통계적으로도 유의미한 차이임을 보였다.

앞서 파이썬 기초 역량과 유사하게 '대화형 프로그래밍' 교과목에선 '인공지능의 이해 및 활용' 교과목보다 각 파트별로 학습 시간도 더 길게 할애되었고, 기초 역량을 높이는데 효과를 보인 수업 활동인 요약과제 역시 동일하게 수행했음에도 평균적으로 유의하게 낮게 평가된 이유는 학습량도 중요하지만, 학습 동기를 높여 학습에 집중토록 하는 것도 필요하기 때문인 듯하다. 문법 난이도가 높아져서 프로그래밍에 어려움을 느끼게 되는 순간에도 '대화형 프로그래밍' 교과목은 문법적 진도에만 집중한 반면, '인공지능의 이해 및 활용' 교과목은 수업 진도 중간에 세미나 시간을 두어 자신의 전공 분야에 왜 파이썬 프로그래밍이 필요한지, 지능형 프로그램이란 무엇인지, 인공지능에서 파이썬의 역할은 무엇인지 등을 학생들에게 스스로 성찰해 보도록 함으로 난이도 높은 학습을 이어갈 학습 동기를 부여하지 않았나 생각되며, 다음의 강의평가 서술형 학생 응답이 이러한 생각을 뒷받침해 준다.

- 세미나 수업으로 전공을 배우기 전에 기초지식을 알 수 있었다.
- 학습할 것과 수행해야 할 과제가 많았던 만큼 많은 것을 배워 성장할 수 있었습니다.
- 프로그래밍이 도움이 되었다.
- 인공지능과 관련되어 실용적인 수업을 진행하였다.
- 세미나 때문에 조금 짝짝하긴 했지만 그렇기 때문에 많이 배운 것 같다.
- 인공지능과 파이썬에 대해 배울 수 있다.
- 인공지능에 대한 이해와 파이썬에 대한 이해가 굉장히 잘 되었다. 기초적인 개론 수업이지만 앞으로 배워나갈 과목들에 대한 흥미를 키우게 되는 계기였다.
- 발표 과제로 인공지능과 프로그래밍의 기초를 모두 배울 수 있다.

IV. 결 론

본 연구에서는 소프트웨어 비전공자의 프로그래밍 기초교육에 있어 학습 목표의 차이에 따라 학습자 교육 만족도와 학습성과의 교육 효과가 차이를 보이는 지를 학생들의 강의평가 결과와 학업 성적을 토대로 분석하였다.

첫째, 교육 만족도의 경우 '인공지능의 이해 및 활용' 교과목은 모든 교육 만족도 항목에서 '대화형 프로그래밍'보다 평균적으로 높은 점수를 기록했다. 특히, 강의 내용의 체계성과 유용성, 교수법의 적합성 등에서 '인공지능의 이해 및 활용'이 두드러진 만족도를 보였다. 이는 '대화형 프로그래밍' 교과목의 '소프트웨어를 활용한 문제 해결을 위한 프로그래밍 역량 강화'란 다소 광범위한 학습 목표보다는 '인공지능의 이해 및 활용'의 '자신의 전공 분야에 활용할 수 있는 지능형 프로그래밍 역량 강화'와 같이 학생들이 이해할 수 있는 구체적인 학습 목표가 교육에 더 효과적이며, 문법에 따른 진도 계획보다

는 인공지능이란 문제 도메인을 학기 초반부터 마지막까지 파이썬 프로그래밍 학습과 함께 인지시키고, 파이썬을 활용한 인공지능 사례에 대한 시연과 실습으로 해결해야 할 문제에 대한 이해도를 높이며 프로그래밍을 함께 학습할 수 있도록 구성된 교과과정과 교수법 때문이라 분석된다.

둘째, 파이썬 프로그래밍 역량에 관한 학습성과 분석 결과를 보면 파이썬 기초 파트에서는 큰 차이가 나타나지 않았으며, 두 교과목의 성취도가 유사했다. 그러나 난이도가 높아지는 파이썬 활용과 자료구조 항목에서는 '인공지능의 이해 및 활용' 교과목이 통계적으로 유의미한 차이를 보이며 더 높은 성취도를 보였다. 이는 '인공지능의 이해 및 활용' 수업에 도입한 학습 활동 중 일반적으로 프로그래밍 교육에선 잘 도입되지 않는 세미나 수업의 영향이 컸던 것으로 분석되었는데, 세미나를 통해 학생들 스스로 전공적 연계성과 프로그래밍의 필요성에 공감하며, 학문적 호기심과 학습 동기를 높일 수 있었고, 그 결과 프로그래밍의 응용 역량 역시 향상시킬 수 있었던 것으로 분석된다.

본 사례 연구는 비전공자를 위한 프로그래밍 교육에도 다양한 교과과정과 교수법을 시도해볼 가치가 있음을 보여주고 있으며 향후 교육 방향 설정에 유용한 지표로 활용될 수 있을 것으로 기대된다.

참고문헌

- [1] Ministry of Education and Ministry of Science, ICT and Future Planning. Promotion Plan of National Center of Excellence in SW [Internet]. Available: <https://www.korea.kr/briefing/pressReleaseView.do?newsId=156066644>
- [2] Korea National Institute for General Education. Standard Model of the Korea National Institute for General Education [Internet]. Available: https://konige.kr/data/general_edu.php.
- [3] National Center of Excellence in SW. Official Website [Internet]. Available: <https://www.swuniv.kr/>.
- [4] B. Oh, J. Lee, and J. Lee, "Exploring Perception and Experience of Non-Majors about SW Education Using CQR," *Journal of the Korean Association of Information Education*, Vol. 23, No. 5, pp. 395-413, October 2019. <https://doi.org/10.14352/jkaie.2019.23.5.395>
- [5] G.-J. Park and Y.-J. Choi, "Exploratory Study on the Direction of Software Education for the Non-Major Undergraduate Students," *Journal of Education & Culture*, Vol. 24, No. 4, pp. 273-292, August 2018. <https://doi.org/10.24159/joec.2018.24.4.273>
- [6] J. E. Nah, "Analysis of Computational Thinking Learning Effect through Learner Observation," *Korean Journal of General Education*, Vol. 11, No. 5, pp. 349-378, October 2017.
- [7] J. E. Nah, "Software Education Needs Analysis in Liberal Arts," *Korean Journal of General Education*, Vol. 11, No. 3, pp. 63-89, June 2017.
- [8] S. 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, May 2015. <https://doi.org/10.32431/kace.2015.18.3.005>
- [9] J.-Y. Seo and S.-H. Shin, "A Case Study on the Effectiveness of Major-Friendly Contents in Software Education for the Non-Majors," *Journal of Digital Convergence*, Vol. 18, No. 5, pp. 55-63, May 2020. <https://doi.org/10.14400/JDC.2020.18.5.055>
- [10] S.-Y. Pi, "A Study on Coding Education of Non-Computer Majors for IT Convergence Education," *Journal of Digital Convergence*, Vol. 14, No. 10, pp. 1-8, October 2016. <https://doi.org/10.14400/JDC.2016.14.10.1>
- [11] W. S. Kim, "A Study on the Recognition of Freshman on Computational Thinking as Essential Course," *Culture and Convergence*, Vol. 39, No. 6, pp. 141-170, December 2017. <https://doi.org/10.33645/cnc.2017.12.39.6.141>
- [12] W. Kim, "Exploring the Direction of Granular Basic-Software Education Considering the Major of College Students," *Journal of the Korean Association of Information Education*, Vol. 23, No. 4, pp. 329-341, August 2019. <https://doi.org/10.14352/jkaie.2019.23.4.329>
- [13] H.-M. Lee, "Case Study of Art Coding Education for Computational Thinking," *Korean Journal of General Education*, Vol. 14, No. 1, pp. 149-166, February 2020.
- [14] E.-S. Seo, "Design of SW Liberal Arts Education to Improve Learning Motivation for Computer Non-Major - Focusing on the Use of Computing Design Think-Up," *Journal of Digital Contents Society*, Vol. 23, No. 11, pp. 2171-2179, November 2022. <https://doi.org/10.9728/dcs.2022.23.11.2171>
- [15] J.-Y. Seo and S.-H. Shin, "A Case Study of Educational Effectiveness by Software Subjects for Humanities College Students," *Journal of the Korea Society of Computer and Information*, Vol. 27, No. 9, pp. 267-277, September 2022. <https://doi.org/10.9708/JKSCI.2022.27.09.267>
- [16] J.-Y. Seo and S.-H. Shin, "Exploring the Effectiveness of Major-Friendly SW Basic Education," *Journal of Digital Contents Society*, Vol. 24, No. 3, pp. 541-549, March 2023. <https://doi.org/10.9728/dcs.2023.24.3.541>
- [17] Ajou University. Software Education Center [Internet]. Available: <https://www.ajou.ac.kr/sw/introduction/busin-ess-introduction.do>.
- [18] Sungkyunkwan University. SW-AI Whole School Education [Internet]. Available: <https://skb.skku.edu/swuniv/vision.do>.

- [19] Yonsei University. SW-Centered University Project Group [Internet]. Available: <https://swuniv.yonsei.ac.kr/>.
- [20] Kyungpook National University. Software Education Institute [Internet]. Available: https://swedu.knu.ac.kr/02_sub/01_sub.html/.
- [21] Chungnam National University. SW Liberal Education [Internet]. Available: <https://swuniv.cnu.ac.kr/swuniv/major/non-major.do/>.
- [22] Hanyang University ERICA. SW Convergence Education [Internet]. Available: <http://computer.hanyang.ac.kr/converge/converge.php/>.
- [23] M. Lee and Y. Kim, "Development of the Unfolding Model of Procedures for the Introductory Programming Education for Non-Majors," *The Journal of Korean Association of Computer Education*, Vol. 23, No. 4, pp. 35-47, July 2020. <https://doi.org/10.32431/kace.2020.23.4.004>
- [24] J. Kim and E. Sohn, "Difficulty Analysis of an Introductory Computer Programming Course for non-Major Students," *Journal of Creative Information Culture*, Vol. 7, No. 2, pp. 69-77, May 2021. <https://doi.org/10.32823/jcic.7.2.202105.69>



구은희(Eun-Hee Goo)

2004년 : 단국대학교
전자컴퓨터공학과(공학석사)
2009년 : 단국대학교
전자컴퓨터공학과
(공학박사-보안 및 암호)

2011년~2013년: 서일대학교 정보통신과 강의전담 교수
2013년~2014년: (주)도넛시스템 LSI 이미징사업부 책임 연구원
2014년~2016년: (주)이너트론 이동통신연구소 수석 연구원
2016년~현재: 아주대학교 다산학부대학 조교수
※관심분야 : 정보보호, 암호 알고리즘, 소프트웨어 교육, 인공지능 교육 등



서주영(Joo-Young Seo)

2001년 : 이화여자대학교
컴퓨터공학과(공학석사)
2009년 : 이화여자대학교
컴퓨터공학과
(공학박사-소프트웨어공학)

1993년~1997년: 삼성전자 시스템LSI사업부 주임연구원
2009년~2009년: 이화여자대학교 컴퓨터공학과 연구교수
2009년~2016년: 아주대학교 소프트웨어학과 강의교수
2016년~현재: 아주대학교 다산학부대학 부교수
※관심분야 : 소프트웨어공학, SW테스트, SW융합교육 등