

앱 인벤터를 활용한 SW프로젝트 교육이 컴퓨팅 사고력 기반 문제해결력과 SW학습 과제가치 인식에 미치는 효과: 인문사회계열 신입생을 대상으로*

The effect of SW project education using App Inventor on the
computational thinking-oriented problem-solving skills and
perceived SW learning task value: For freshmen in the humanities
and social sciences

유지원[†]
Ji Won You[†]

요 약

대학에서 비전공자까지 SW교육이 확대되고 있는 추세 가운데, 본 연구는 인문사회계열 대학 신입생을 대상으로 앱 인벤터(App Inventor)를 활용한 소프트웨어(SW) 프로젝트 수업의 효과성을 검증하였다. 컴퓨팅 사고력 기반 문제해결력과 SW학습 과제가치 인식을 학습성과 변인으로 선정하여 사전-사후 점수를 비교하였고, 학습자가 지닌 SW학습 과제가치 수준에 따라 컴퓨팅 사고력 기반 문제해결력과 학습만족도에 차이가 있는지 분석하였다. 자료수집은 경기도 A대학에서 해당 강좌를 수강한 인문사회계열 학생 중 연구참여에 자발적으로 응답한 1,010명으로부터 이루어졌다. t-test 분석 결과, 학습자의 사전-사후 컴퓨팅 사고력 기반 문제해결력과 SW학습 과제가치 인식이 유의하게 향상되었으며, SW학습 과제가치 인식 수준에 따라 구분된 집단 간에 컴퓨팅 사고력 기반 문제해결력과 학습만족도에 차이를 보였다. 연구결과를 토대로 비전공자의 특성을 고려하면서 실제적인 문제해결 경험을 제공할 수 있는 SW교육을 설계하고, 정의적인 요소를 교육성으로 포함할 것을 제안하였다.

주제어: 소프트웨어 교육, 앱 인벤터, 컴퓨팅 사고력 기반 문제해결력, SW학습 과제가치, 인문사회계열

ABSTRACT

In the trend of expanding software (SW) education from university to non-major students, this study was conducted to examine the effectiveness of SW project courses using App Inventor for freshmen in humanities and social sciences. Computational thinking-oriented problem-solving skills and perceived SW learning task value were selected as learning outcomes and the pre-post scores were compared. Also, it was analyzed whether there is a difference in computational thinking-oriented problem-solving skills and learning satisfaction according to the level of SW learning task value. Data were collected from 1,010 students who took the courses and voluntarily participated in the research among the humanities and social sciences students at A University in Gyeonggi-do. As a result of t-test analysis, the post-test score of computational thinking-oriented problem-solving skills and the post-test score of the perceived SW learning task were significantly improved compared to each prior score. Moreover, there were significant differences between the two groups which were divided according to the level of perception of SW learning task value in computational thinking-oriented problem-solving skills and learning satisfaction. Based on the research results, it was proposed to design SW education that can provide authentic problem-solving experience while considering the characteristics of non-major students, and to include affective factors as educational outcomes.

Keywords: Software education, App Inventor, computational thinking-oriented problem solving skills, task value of SW education, humanity and social science students

[†]정 회 원: 가천대학교 교육대학원 AI융합교육전공 부교수

논문투고: 2022년 08월 11일, 심사완료: 2022년 09월 23일, 게재확정: 2022년 09월 23일

* 이 논문은 2022년도 가천대학교 교내연구비 지원에 의한 결과임.(GCU-202207190001)

1. 서론

현대 사회는 빅데이터, 인공지능, 사물인터넷(IoT) 기술을 중심으로 다양한 영역에서 초연결과 융합이 진행되고 있다. 교육현장에서도 디지털 사회에 대응할 수 있는 문제해결력과 사고력을 지닌 창의융합인재 양성에 힘쓰고 있으며, 소프트웨어(SW)가 새로운 가치 창출을 이끄는 기조로 기여한다. SW중심사회에서는 SW가 더 이상 전공자만을 위한 학문이 아니며, 아이디어를 SW로 구현하거나 활용하여 문제를 해결하는 강력한 도구로써 모든 학생들이 배워야 할 대상이다[1-3]. 따라서 미래 사회의 필수 역량으로써 SW 교육 방향은 학습자의 컴퓨팅 사고력(computational thinking) 증진에 초점을 둔 문제해결 능력을 키우는 SW교육으로 설정하고 있다[4]. 컴퓨팅 사고력이란 컴퓨팅 시스템을 활용하여 문제를 효과적으로 해결할 수 있도록 표현하는 사고과정으로[5] 문제해결과정으로 이해할 수 있다[6].

SW 역량이 중요해짐에 따라 다수 대학에서 비전공자까지 대상으로 SW교육을 확대하고 있다. 하지만 종종 전공자와의 차이점과 컴퓨팅 사고력 교육 의도를 충분히 이해하지 못한 채 프로그래밍 기능 위주의 교육이 이루어진다고 지적된다[1, 7, 8]. 비전공자의 경우, 낮은 프로그래밍 언어의 구문을 습득하는 것과 컴퓨터과학에 대한 개념을 이해하는 것에 어려움을 표출하고, 프로그래밍 교육의 가치와 필요성을 인식하지 못하거나 프로그래밍 과정을 지루하고 어렵게 느낀다[9-11]. 그런데 컴퓨팅 사고력 교육은 코딩 기술 이상의 것으로, 각 학문 영역 및 실제 생활과 관련된 문제를 컴퓨팅 사고로 해결하는 고차적인 접근이 필요하다[1]. 또한 SW교육에서는 인지적 능력 못지 않게 학습자의 정서와 동기 등 정의적 요소가 중요하다는 점을 간과해서는 안 된다. 복잡하고 불확실성이 높은 문제를 해결하기 위해서는 실패를 극복하면서 학습자가 끈기있게 노력하는 것이 중요하며, 학습자의 동기 요소는 미래의 지속적인 SW학습을 결정하는데 영향을 미치기 때문이다[6, 12].

최근 대학에서는 이러한 문제점을 개선하기 위해 계열별 맞춤 SW교육과정을 개발·운영하는 노력을 시도하거나[7], 비전공자들의 흥미를 높이면서 인지적 부담을 줄일 수 있는 방안과 창의력과 문제해결력을 함께 함양할 수 있는 교육과정을 적용하고 있다. 이러한 측면에서 블록형 프로그래밍을 활용한 SW교육은 문법적인 구문 오류 없이 블록을 조립하는 방식

으로 쉽게 프로그래밍 할 수 있어 학습자가 문제해결에 집중하고 결과물을 만들면서 자연스럽게 흥미를 느낄 수 있는 장점이 있다. 특히 앱 인벤터(App Inventor)는 블록형 프로그래밍으로 스마트 폰 앱을 구현할 수 있는 언어이다. 학습자가 개발 과정에서 즉각적으로 결과를 확인할 수 있어 지루하지 않게 문제해결을 경험할 수 있어 입문자 또는 비전공자 대상 SW교육 도구로 추천된다[13]. 또한 효과적인 컴퓨팅 사고력 교육은 일상에서 문제를 발견하고 컴퓨팅시스템으로 문제를 스스로 해결해나가는 과정에서 자연스럽게 훈련될 수 있기 때문에 앱 인벤터를 활용하여 문제해결 학습이 가능하다. Czerkawski와 Lyman(2015)은 컴퓨팅 사고력 교육이란 궁극적으로 학습자가 효과적인 문제해결자가 되도록 하는 것이라고 강조한 바 있다[14]. 특히 실생활에서 문제를 발견하는 것은 학습자가 주제와의 관련성을 높이 인식하여 학습흥미와 몰입을 촉진할 수 있다는 점에서 주목할 만하다[2]. 이러한 맥락에서 앱 인벤터를 활용한 프로젝트형 수업이 컴퓨팅 사고를 통한 문제해결 경험을 제공하는 동시에 학습자의 동기도 촉진할 수 있을 것으로 사료된다.

본 연구는 대학의 인문사회계열 신입생을 대상으로 운영된 SW교과목에서 앱 인벤터를 활용한 SW프로젝트 교육을 통해 학습자의 컴퓨팅 사고력 기반 문제해결력과 SW학습에 대한 가치 인식에 변화가 있는지를 검증하고자 하였다. 나아가 SW학습에서 학습자의 정의적 요인이 중요한 만큼, SW학습에 대한 가치를 인식하는 수준에 따라 SW교육성과에 차이를 보이는지 살펴보고자 하였다. 구체적인 연구문제는 다음과 같다.

연구문제 1: 앱 인벤터를 활용한 SW프로젝트 교육은 인문사회계열 학습자의 컴퓨팅 사고력 기반 문제해결력 향상에 효과적인가?

연구문제 2: 앱 인벤터를 활용한 SW프로젝트 교육은 인문사회계열 학습자의 SW학습 과제가치 인식 향상에 효과적인가?

연구문제 3: 인문사회계열 학습자가 인식한 SW학습 과제가치 수준에 따라 컴퓨팅 사고력 기반 문제해결력에 차이가 있는가?

연구문제 4: 인문사회계열 학습자가 인식한 SW학습 과제가치 수준에 따라 앱 인벤터를 활용한 SW프로젝트 교육에 대한 학습만족도에 차이가 있는가?

2. 이론적 배경

2.1 프로젝트 기반 SW교육

우리나라는 초·중·등 2015 개정 교육과정에서 SW교육을 의무화하면서 학생들의 SW역량 강화에 힘쓰고 있다[15]. 뿐만 아니라 대학의 경우 소프트웨어중심대학 사업이 촉매가 되어 다수 대학에서 전교생을 대상으로 SW교육을 확대하고 있다. 2015년부터 추진된 소프트웨어중심대학 사업은 SW 전문인력을 양성하고 비전공자도 SW소양을 갖추도록 융합 인재가 되도록 교육하여 SW가치를 확산하고 SW경쟁력을 높이는 데 목표를 두고 있다. 다수 대학이 소프트웨어중심대학 사업에 참여하고 있으며, 융합교육이나 교양교육 과정으로 소프트웨어 교육과정을 개설하고 운영하고 있다. 이러한 SW교육의 방향은 학생들의 컴퓨팅 사고력을 향상시키고 컴퓨팅 시스템을 활용하여 문제해결 능력을 함양하는 데 초점을 두고 있다[7, 8, 16].

대학에서 SW교육이 보편적 교육으로 활성화되는 가운데, 홍성연 외(2021)는 SW 기초교육을 수강한 학습자가 SW문해력과 컴퓨팅 사고력 향상에 큰 효과를 보였고[8], 박주연(2021)은 SW교육 전에는 계열별로 컴퓨팅 효능감에 차이가 있었으나 교육 후 계열별 차이가 사라졌다고 논의하면서 SW교육의 효과를 입증하였다[1]. 한편, 비전공자들의 SW교육에 대한 우려도 지적되고 있다. 비전공자들은 고등학교에서의 수학 및 과학 등의 수강 과목에 차이로 관련 배경지식이 다르고[9], 자기 전공과의 관련성을 쉽게 인식하기 어려우며, 미래 활용도에 대한 기대가 낮아 학습자의 흥미와 동기 등에 전공자에 비해 낮다고 논의한다[10]. 또한 비전공자들은 프로그래밍 언어의 구문을 작성하고 해석하는 등의 어려움과 오류를 많이 범하고, 변수, 리스트, 명령어 선택 등 설계와 알고리즘에 대한 이해와 문제 해결을 어렵힌다고 논의된다[9]. 학자들은 비전공자의 경우 SW학습에 대한 필요성을 인식하는 것이 최우선 과제라고 제언하고 있다[10, 17].

이에 비전공자들의 특징과 전공 특성을 고려한 컴퓨팅 사고력 교육으로 학습자 스스로 문제를 탐색하고 발견할 수 있는 기회를 제공하고 동료와의 상호작용과 협업으로 학습하는 구성주의적 접근할 것을 제안하고 있다[9, 17]. 특히 실제 생활에서 문제를 발견하고 이를 해결해나가는 프로젝트 기반 학습으로 컴퓨팅 사고력을 효과적으로 학습할 수 있다고 논의된다[9, 17, 18]. 특히 전통적인 교육방식으로 프로그래

밍에 대한 지식과 기술 중심 학습에서 탈피하고, 학습자 스스로가 프로젝트를 만들고 필요한 지식과 기술을 찾아 배우는 경험이 필요하다고 강조한다[18]. 실제 세계와 유리된 채 ‘변수’, ‘반복’, ‘조건문’ 등의 프로그래밍 구문을 배우는 것은 학습자가 왜 컴퓨터 과학을 배워야 하는지를 알기 어렵고, 향후 어떤 상황에 이것을 쓸 수 있는지 조망할 수 없기 때문이다[19, 20].

실제 연구에서 비전공자에 해당하는 예비교사를 대상으로 한 연구에서는 프로젝트 수행 경험이 컴퓨팅 사고력 함양에 중요한 요인으로 나타났다[21], 비이공계열 대학생들이 참여한 해커톤(hackathon) 활동 이후 사전에 비해 문제해결능력이 향상되었다[22]. 또한 성정숙, 김수환, 김현철(2015)은 비전공자들이 학습난이도가 상승하면서 SW학습에 어려움을 많이 느끼지만, 프로젝트를 수행한 후 성취감이 높아져 SW학습에 대한 흥미가 향상된다고 밝히면서 프로젝트 기반 학습을 지지하였다[23].

2.2 컴퓨팅 사고력과 컴퓨팅 사고력 기반 문제해결력

컴퓨팅 사고력이란 컴퓨팅 시스템을 활용하여 문제 발견, 문제 이해, 해결 방향 설정, 해결안을 개발하여 효과적으로 문제를 해결하는 절차적 사고 능력을 뜻한다[24]. 컴퓨팅 사고력의 개념을 대중화시킨 Wing(2006)은 컴퓨팅 사고력이 컴퓨터과학자에게만 요구되는 것이 아니라 디지털 정보화 시대를 살아가는 모든 사람들에게 필요한 역량이라고 주장한다[5]. ISTE(International Society for Technology in Education)와 CSTA(Computer Science Teachers Association)(2011)에 따르면, 컴퓨팅 사고력의 구성요소를 자료 수집(data collection), 자료 분석(data analysis), 자료 표현(data presentation), 문제 분해(problem decomposition), 추상화(abstraction), 알고리즘과 절차(algorithms & procedure), 자동화(automation), 시뮬레이션(simulation), 병렬화(parallelization) 9개 요소로 정의하였다[25].

디지털 정보사회에서 컴퓨팅 사고력 기반 문제해결력은 사람들에게 범용적으로 필요한 사고력이다[4]. 컴퓨팅 사고력 기반 문제해결력이란 컴퓨팅 시스템 활용을 전제로 인간의 사고를 프로그램으로 구현하여 복잡한 문제를 해결하는 것을 뜻한다. 이정훈(2020)은 컴퓨팅 사고력 기반 문제해결과정을 문제인식, 문제

분석, 해결책 구상, 알고리즘 설계, 프로그램 구현, 공유 및 평가에 이르는 6단계로 구분하고, 각 단계별로 관련된 컴퓨팅 사고력 요소인 추상화, 자료수집, 자료 표현, 구조화, 자동화, 일반화, 평가를 연결하여 컴퓨팅 사고력 기반 문제해결력 평가 모델과 평가 도구를 개발하였다[26]. 유사하게 이민우와 김성식(2019)은 문제분해(문제 이해, 분석, 분할), 추상화(패턴인식, 핵심요소추출, 모델링), 알고리즘 절차(알고리즘 설계, 표현, 분석), 자동화(프로그래밍, 자료구조, 디버깅)로 개념화하여 평가 문항을 개발하였다[27].

2.3 SW학습 과제가치

SW학습 과제가치란 학습자가 SW학습에 대해 가지는 흥미, 유용성, 중요성 등에 대한 인식으로 학습자의 동기 및 태도를 나타내는 요인이다[12]. 일반적으로 과제가치를 높이 인식할수록 학습자는 인내심을 가지고 도전적인 과제를 선택하며, 내재적인 동기와 인지전략 사용을 촉진한다고 알려져 있다[28, 29].

SW교육 관련 선행연구에서 SW교육에 대한 흥미와 태도는 학습의 목표이자 학습성파에 주요 요인으로 나타난다[1, 8, 12]. SW학습에 대한 가치를 인식하고 흥미가 높을수록 프로그래밍으로 복잡한 문제를 해결하는데 노력을 기울이고, 새로운 해결안을 지속적으로 시도하며, 목표를 성취하기 위해 끈기있게 학습하는 등 학습성파와 학습지속의향에 긍정적 영향을 미친다[12, 30]. 김수환(2015)은 SW 비전공자를 대상으로 한 SW교육이 학습에 대한 가치와 동기, 흥미를 인식하지 못할 경우 오히려 컴퓨팅 사고력 육성에 부정적일 수 있음을 우려하였고[9], 박윤수와 이민정(2020)은 비전공자들의 SW교육에 대한 부정적 인식을 경감시킬 수 있도록 실생활 문제를 활용하고 교수자와의 소통을 강조하는 교육과정 설계를 제안하였다[17]. 박주연(2021)은 대학생의 컴퓨팅 흥미 점수는 인문사회 계열이 공학 계열 학생들에 비해 유의하게 낮았으나, 컴퓨팅 사고력 교양 교과목 수강 후에는 두 계열 간 차이가 사라져 컴퓨팅 사고력 교육의 효과를 입증하였다. 또한 컴퓨팅 흥미 수준에 따라 구분된 상하 집단 간에 융합적 사고력, 융합적 가치창출, 컴퓨팅 효능감이 유의한 차이를 보여 SW교육에서 학습자의 정의적 요소의 중요성을 강조하였다[1]. 종합하면, SW학습에 대한 과제가치 인식과 SW학습성과 및 노력 간에 밀접한 관계를 가지고 있음을 시사한다.

2.4 앱 인벤터 활용 교육과 효과

앱 인벤터는 복잡한 프로그래밍이 없이도 아이디어를 표현하고 스마트폰 앱을 구현할 수 있는 블록 기반의 프로그래밍 언어이다. 코드 블록을 드래그 앤 드롭으로 쉽게 프로그래밍 할 수 있는 환경을 제공하고, 프로그램이 시각적으로 구조화되어 설계 내용을 쉽게 이해할 수 있다. 앱 인벤터는 애플리케이션 디자인 뷰(application design view)와 블록 에디터(blocks editor)의 작업 영역으로 구성되어 있으며, 블록 에디터에서 코드를 선택하고 조합하여 원하는 동작을 구현하는 프로그래밍을 할 수 있다. 또한 스크래치와 달리 앱 인벤터는 모바일 컴퓨팅을 전제로 하고, 구현한 코드를 스마트폰에서 즉시 확인하면서 프로그래밍 할 수 있다는 점이 큰 장점으로 꼽힌다. 이러한 특징들로 인해 앱 인벤터는 SW교육용 언어로 190개국 이상에서 널리 사용된다[13, 20, 31, 32].

앱 인벤터 활용 교육에 대한 선행연구를 살펴보면, 앱 인벤터를 활용 교육을 통해 학습자들의 프로그래밍에 대한 흥미와 자신감이 향상되었으며[31, 33], 프로그래밍 학습에 대한 동기가 향상되었다[34]. 또한 앱 인벤터를 활용한 SW교육은 학습자의 컴퓨팅 사고력을 촉진하고[35], 창의적 문제해결력을 향상시키는데 기여하였다[3, 35]. Deng(2017)은 학생들이 앱 인벤터를 사용하여 실시간으로 협력하는 프로그래밍 학습을 통해 학생들 간 상호작용이 향상되고, 협력적 프로그래밍에 대한 선호도가 증가하였다고 규명하면서 앱 인벤터를 협력적 프로그래밍 교육 도구로 잠재성을 논의하였다[36]. 이러한 연구결과를 종합해 볼 때, 앱 인벤터는 학습자가 전문적인 프로그래밍 없이 문제해결에 집중하면서 컴퓨팅 사고력을 함양하는데 효과적이며 SW교육에 대한 흥미를 증진하는데 긍정적으로 작용함을 알 수 있다.

3. 연구방법

3.1 연구대상

본 연구의 자료수집은 수도권 A종합대학에서 신입생 교양교과목으로 개설된 SW프로젝트 수업을 표집하고, 2022-1학기에 해당 과목을 수강한 인문사회계열 대학생으로부터 이루어졌다. 연구 참여에 자발적으로 동의하고, SW프로젝트 교육 사전-사후에 진행된 2회의 설문조사에 모두 참여한 1,010 사례를 분석에 활용하였다. 남학생은 422명(41.8%), 여학생은

588(58.2%)명이었으며, 경영학과, 심리학과, 사회복지학과, 경제학과, 미디어커뮤니케이션 학과, 행정학과, 법학과, 영미어문학과 등 인문사회계열에 속한 다양한 전공의 학생들이 참여하였다. 그러나 타계열 학생의 자료 수집은 이루어지지 못해 계열 간 교육 효과성은 비교하지는 못하였다.

3.2 측정도구

연구변인의 측정은 모두 5점 척도로 구성된 설문문항으로 이루어졌다. 설문도구는 교육학 전문가 3인의 검토를 거쳤다. 각 변인별 설명은 다음과 같으며, Table 1에 요약하여 제시하였다.

첫째, SW학습 과정가치 측정을 위해서 Artino와 McCoach(2008)의 과정가치 인식 도구를 SW교육 맥락에 맞게 수정하였다[37]. 총 4문항을 활용하였고, 문항 예는 “프로그래밍, 소프트웨어 학습/활동은 시간과 노력을 들일 가치가 있다” 등이다. SW학습 과정가치 도구의 사전 신뢰도는 .83, 사후 신뢰도는 .90이었다.

둘째, 컴퓨팅 사고력 기반 문제해결력 측정은 이정훈(2020)의 도구를 참고하여 활용하였다[26]. 이정훈(2020)에서 문제해결과정 중심 컴퓨팅 사고력에 포함된 요소는 문제인식, 추상화, 자료수집, 자료분석, 자동화, 공유 및 평가이다. 측정에는 총 12문항이 사용되었고, 문항 예는 “유사한 문제해결 과정을 탐색하여 반복되는 패턴이나 핵심 요소를 찾는다” 등이다. 컴퓨팅 사고력 기반 문제해결력의 사전 신뢰도는 .89, 사후 신뢰도는 .92로 높게 나타났다.

셋째, SW학습에 대한 학습만족도는 4문항으로 정세삼과 임규연(2000)의 팀 학습 만족도 측정도구를 연구맥락에 맞게 수정하여 활용하였다[38]. 문항 예는 “팀 활동을 통한 SW 프로젝트 수행은 좋은 경험이었다” 등이었고, 신뢰도는 .89로 나타났다.

Table 1. Summary of research instruments

variable	sample item	number of items	Cronbach's α
Task value in SW education	It is worth investing time and effort into programming and software learning activities.	4	.83(pre), .90(post)
CT oriented problem solving skill	Explore similar problem-solving processes to find recurring patterns or key elements.	12	.89(pre), .92(post)
Learning satisfaction	It was a good experience to be involved in a team-based SW project.	4	.89

3.3 연구절차

연구가 수행된 대학은 실생활에서의 문제를 해결할 수 있는 창의적인 아이디어를 제안하고 프로그래밍으로 구현하는 SW 프로젝트를 교양교육과정으로 운영하고 있다. 해당 과목은 신입생들의 필수교양과목으로 운영되며, 2022-1학기 과정은 WebEx 실시간화상 강의 플랫폼으로 온라인으로 진행되었다. 해당 교과목은 1학점으로 총 15차시로 개발되었고, 집중하기로 진행되어 학생들은 월-금요일까지 5일간 매일 3시간 이상 수업과 팀 프로젝트 활동에 참여하였다. 인문사회계열 신입생을 대상으로 한 교육내용은 1주차(1-3차시)에는 프로그램에 대한 교육에 대한 오리엔테이션, 팀 빌딩 활동, MIT 앱 인벤터 사용에 대한 기본 강의를 진행하고 팀별로 학생들은 각자 역할을 논의하면서 프로젝트 수행을 위한 준비를 한다. 2주차(4-6차시)에는 디자인 씽킹에 대한 강의를 실시하고, 이어서 각 팀별로 주어진 대주제에 대해 아이디어를 탐색하면서 팀별 세부 주제를 논의하여 결정한다. 각 팀은 활동 결과물로 프로젝트에 대한 해결 아이디어를 작성하여 제출한다. 3주차(7-9차시)는 팀별 프로젝트 수행 시간으로 각 팀은 선정한 주제에 대하여 팀원과 협력하면서 아이디어를 앱 인벤터로 구현한다. 4주차(10-12차시)는 팀별로 프로그래밍을 완료하여 프로젝트를 마무리하는 시간으로, 팀원들은 프로그래밍 결과를 지속적으로 평가하고 보완하면서 산출물을 완성한다. 5주차(13-15차시)에는 팀별 발표와 평가로 진행되며, 복수의 심사위원이 평가를 통해 우수작을 선정한다. 학생들의 몰입과 동기 향상을 위해 우수팀에게는 소정의 장학금이 수여된다. 해당 학기에 제시된 대주제는 사회 구성원의 행복 증진(복지, 치안, 사회적 약자 배려 등)에 도움을 줄 수 있는 앱, 대학생활을 즐겁게 할 수 있는 앱 개발, 일상생활의 불편함을 해소해 줄 수 있는 앱 등이었다. 연구를 위한 설문조사는 교과목 시작 전과 종료 직후에 모바일로 실시하였다.

3.4 자료분석

수집된 자료는 Jamovi로 분석하였다[39, 40]. 연구변인의 사전, 사후 측정값의 기술통계, 신뢰도 분석을 실시하고, SW학습 과정가치 인식과 컴퓨팅 사고력 기반 문제해결력 변인별로 사전-사후 차이 분석은 대응표본 *t*-test를 수행하였다. 사전 SW학습 과정가치 수준에 따라 두 집단으로 구분하고, SW학습 과정가치 상하 집단 간에 컴퓨팅 사고력 기반 문제해결력과 학

습만족도에 차이가 있는지를 독립표본 *t*-test로 분석하였다. Cohen's *d*값으로 효과크기를 산출하였다. Cohen's *d*는 .20일 경우 작은 효과, .50은 중간크기의 효과, .80이상은 큰 효과로 해석한다[41]. 가설검정에 사용한 유의수준은 .05이다.

4. 연구결과

4.1 기술통계

본 연구의 SW학습 과제가치, 컴퓨팅 사고력 기반 문제해결력 변인의 사전, 사후 측정된 값의 기술통계 분석 결과는 Table 2와 같다. 사전 SW학습 과제가치의 평균은 3.03이었는데, 이는 인문사회계열 학생들의 사전 과제가치 인식이 높지 않음을 보여주었다. 사후 평균은 4.40이었다. 사전 측정된 컴퓨팅 사고력 기반 문제해결능력의 평균은 3.43, 사후 평균은 3.97로 나타났다. 사후 측정된 학습만족도의 평균은 4.62로 높게 나타나 학습자들이 전반적으로 운영된 SW교육에 만족한 것으로 해석할 수 있다. 연구변인 간 상관은 .10-.56 범위에서 정적인 관계를 보였다.

Table 2. Descriptive statistics and correlational analysis

variables	1	2	3	4	5
1. Pre task value in SW education	1.00				
2. Post task value in SW education	.22*	1.00			
3. Pre CT oriented problem solving skill (Pre CT-PBS)	.50*	.26*	1.00		
4. Post CT oriented problem solving skill (Post CT-PBS)	.43*	.47*	.56*	1.00	
5. Learning satisfaction	.10*	.63*	.19*	.34*	1.00
<i>M</i>	3.03	4.40	3.43	3.97	4.62
<i>SD</i>	.83	.66	.64	.62	.51
Skewness	-.01	-1.00	-.23	-.36	-1.30
Kurtosis	-.15	.78	.45	.35	1.34

**p* < .05

4.2 SW학습 과제가치에 대한 사전-사후 차이 분석

교육 전 측정한 연구 참여자가 인식한 SW학습 과제가치 점수와 사후 점수 간에 통계적으로 유의한 차이가 있는지를 대응표본 *t*-test로 검증하였다(Table 3 참조). 분석결과, 사후-사전 점수 차이는 1.37이었으며, *t* 통계값은 46.43(*p* < .001)으로 나타나 앱 인벤터를

활용한 SW프로젝트 교육을 통해 학습자가 인식하는 SW학습 과제가치가 유의하게 향상된 것으로 확인하였다. Cohen's *d*=1.46으로 큰 효과크기로 해석할 수 있다. 이는 비전공자들도 SW학습이 자신들과의 무관한 분야가 아님을 인식하고, SW가 실생활의 문제해결에 유용하게 활용될 수 있음을 직접 체험함으로써 SW과제가치 인식 향상에 기여했을 것으로 사료된다.

Table 3. Paired t-test results of pre and post task value in SW education

	<i>M</i> (<i>SD</i>)	Mean diff. (Post-Pre)	<i>t</i>	<i>p</i>	Cohen's <i>d</i>
Pre task value	3.03 (.83)	1.37	46.43***	<.001	1.46
Post task value	4.40 (.66)				

****p* < .001

4.3 컴퓨팅 사고력 기반 문제해결력에 대한 사전-사후 차이 분석

연구 참여자의 사전 컴퓨팅 사고력 기반 문제해결력과 사후 컴퓨팅 사고력 기반 문제해결력 간에 통계적으로 유의한 차이가 있는지를 검정하고자 대응표본 *t*-test를 실시하였다(Table 4 참조). 분석결과, 사후-사전 점수 차이는 .54이었으며, *t* 통계값은 29.06(*p* < .001)로 나타나 사전에 비해 사후 컴퓨팅 사고력 기반 문제해결력이 유의하게 향상된 것을 알 수 있다. 이 차이는 Cohen's *d*=.91로 큰 효과크기로 해석할 수 있다. 이 결과는 SW교육을 지식과 기능 위주가 아니라 실제 문제를 발굴하고 분석과 아이디어 탐색을 통해 문제해결을 시도한 디자인씽킹 기반의 프로젝트 방식이 문제해결력 향상에 기여한 것으로 보인다. 특히 아이디어를 논리적인 알고리즘으로 구현하는 경험과 앱 인벤터로 결과를 즉각적으로 확인하고 오류를 스스로 수정해나가는 반복적이고 성찰적인 과정에서 컴퓨팅 사고력과 문제해결력이 향상된 것으로 사료된다.

Table 4. Paired t-test results of pre and post CT-PBS

	<i>M</i> (<i>SD</i>)	Mean diff. (Post-Pre)	<i>t</i>	<i>p</i>	Cohen's <i>d</i>
Pre CT-PBS	3.43 (.64)	.54	29.06***	<.001	.91
Post CT-PBS	3.97 (.62)				

****p* < .001

4.4 SW학습 과제가치 인식 수준에 따른 사후 컴퓨팅 사고력 기반 문제해결력 차이 분석

연구 참여자가 교육 전 인식한 SW학습 과제가치의 중앙값(median=3.00)을 기준으로 상하 집단으로 구분하였다. 두 집단 간에 컴퓨팅 사고력 기반 문제해결력에 차이가 있는지 검증하였다. 독립표본 *t*-test로 분석 결과, Table 5와 같이, 과제가치 인식이 낮은 집단(n=418)의 컴퓨팅 사고력 기반 문제해결력의 평균은 3.71, 높은 집단(n=592)의 평균은 4.15로 나타나 두 집단 간에 유의한 차이가 나타났다($t = 11.98, p < .001$). 이 차이는 Cohen's $d = .77$ 로 큰 효과크기로 해석할 수 있다. 이는 SW학습에 대한 가치를 인식하는 것이 SW교육에서 중요한 동기적 요소일 뿐 아니라 교육의 성과인 컴퓨팅 사고력 기반 문제해결력과도 밀접하게 연관됨을 시사하는 결과로 해석할 수 있다.

Table 5. Independent *t*-test results of post CT-PBS between two groups

	<i>M</i> (<i>SD</i>)	Mean diff. (High-Low)	<i>t</i>	<i>p</i>	Cohen's <i>d</i>
Low pre task value (n=418)	3.71 (.62)	.44	11.98***	<.001	.77
High pre task value (n=592)	4.15 (.55)				

*** $p < .001$

4.5 SW학습 과제가치 인식 수준에 따른 학습만족도 차이 분석

SW학습 과제가치의 사전 인식 수준에 따라 구분된 두 집단 간에 학습만족도에 차이가 있는지 분석하였다. 독립표본 *t*-test를 실시한 결과 Table 6과 같이, 두 집단 간에 유의하게 차이가 있었으며($t = 2.35, p = .02$) 효과성 크기는 Cohen's $d = .15$ 로 나타나 작은 효과크기로 해석할 수 있다. 인문사회계열 학습자에게 SW학습은 전공이 아니므로 SW학습에 대한 필요성, 중요성, 유용성 등의 가치 인식이 없이는 SW학습에 몰입하기 어렵다. 또한 SW 프로젝트를 수행하는 것은 도전적인 과제로, SW학습 과제가치 인식이 높은 집단에서 SW학습만족도가 더 유의하게 높은 것으로 해석할 수 있다.

Table 6. Independent *t*-test results of learning satisfaction between two groups

	<i>M</i> (<i>SD</i>)	Mean diff. (High-Low)	<i>t</i>	<i>p</i>	Cohen's <i>d</i>
Low pre task value (n=418)	4.57 (.54)	.08	2.35*	.02	.15
High pre task value (n=592)	4.65 (.49)				

* $p < .05$

5. 논의 및 결론

본 연구는 인문사회계열 신입생을 대상으로 앱 인벤터를 활용한 SW프로젝트 교육을 실시하고 컴퓨팅 사고력 기반 문제해결력과 SW학습 과제가치에 대한 효과를 검증하였다. 연구의 주요 결과 및 논의는 다음과 같다. 첫째, 앱 인벤터를 활용한 SW프로젝트 교육에 참여한 학생들의 사전 점수에 비해 사후에 측정된 SW학습 과제가치와 컴퓨팅 사고력 기반 문제해결력이 통계적으로 유의하게 향상되었다. 이는 앱 인벤터와 같은 블록 프로그래밍 교육 및 SW프로젝트 교육으로 컴퓨팅 사고력과 흥미가 증진된다는 연구 등을 뒷받침하는 결과이다[13, 31, 33, 35]. 이 결과는 연구를 수행한 대학에서 SW프로젝트 교육으로 추구한 인지적 성과와 정의적 성과를 모두 달성한 것으로 해석할 수 있다. 특히 분석에서 사전 측정된 SW학습 과제가치가 높지 않아 인문사회계열 학생들의 특성을 보여준 한편, 교육 후에는 4.40으로 괄목할만하게 향상되면서 큰 효과크기를 보였다. 이는 교육에 활용한 앱 인벤터가 비전공자들의 인지적 부하를 줄이면서 쉽게 배워 즉시 문제해결에 응용할 수 있으며, 팀원들과 함께 도전적 프로젝트를 수행한 경험이 학생들의 SW학습에 대한 두려움과 저항감을 경감시키고, SW가 삶에 용한 도구임을 지각하는 데 도움이 되었을 것으로 생각할 수 있다. 종합해 볼 때, 앱 인벤터를 활용한 SW프로젝트 학습이 인문사회계열 학생들의 SW교육에 적절하였다고 판단할 수 있다.

둘째, SW학습 과제가치 인식 수준에 따라 상하 집단으로 구분하고, 교육 후 컴퓨팅 사고력 기반 문제해결력과 학습만족도에 차이가 있는지 검증하였다. 그 결과, SW학습에 대한 과제가치가 높은 집단에서 교육 후에 측정된 컴퓨팅 사고력 기반 문제해결력과 학습만족도가 모두 유의하게 높게 나타났다. 이 결과는 SW학습에 대한 흥미 및 가치 인식이 컴퓨팅 사고력

향상에 중요하다고 논의한 선행연구와 일맥상통한 결과로[1, 9] SW교육에서 정의적 요소의 중요성을 시사하였다.

본 연구결과에 따른 시사점은 다음과 같다. 첫째, 학습자가 지닌 특징, 전공배경, 숙련도를 고려하면서 SW학습에 대한 가치에 공감하고 도전적이고 실제 문제해결의 학습경험을 이끄는 SW 교육을 설계할 필요가 있다. 연구에 활용된 교과는 해당 대학에서 신입생 비전공자들을 대상으로 설계된 교과로 교육도구로 앱 인벤터를 선택한 것은 입문자들의 SW학습에 대한 진입장벽을 낮추고자 한 것이었다. 또한 강의를 최소화하고 팀 프로젝트를 선택한 것은 동료 학습자와 협업으로 문제를 해결하면서 자기주도적인 학습태도와 협력적 문제해결역량을 증진하고자 하는 의도도 가지고 있다. Papert(2000)는 컴퓨팅 사고력 교육이 컴퓨팅 개발자 또는 전문가들이 공개 커뮤니티 안에서 서로 문제를 공유하고 협력적 지성으로 문제를 해결하는 방식이 어울린다고 설명하면서 배우는 방식과 활용하는 방식의 간극을 줄일 것을 주장하였다[18]. 또한 미래사회역량으로써 컴퓨팅 사고력 교육의 궁극적인 목적은 실제 세계의 복잡한 문제를 다각도로 바라보고 다양한 분야를 접목하는 문제해결력 증진임을 염두에 둔다면, SW교육이 현실의 문제를 다루는 프로젝트 중심으로 학습자에게 풍부한 참 경험(rich and authentic experience)을 제공할 수 있도록 설계되어야 한다.

둘째, 비전공자의 SW학습에 대한 흥미와 동기, 가치 인식, 진로 열망 등과 같은 정의적 요소가 SW교육에서 중요 평가 지표로 관리될 필요가 있다. 컴퓨팅 사고력, 컴퓨팅 사고력 기반 문제해결력, 융합적 사고력 등의 인지적 성과는 SW교육에서 주로 관리되는 교육성과 지표이다. 그런데, 비전공자의 경우, 자칫 SW는 자신과 상관없는 분야라고 인식하기 쉽고, 왜 배워야 하는지에 대한 이유를 알지 못할 수 있다. 역량을 발휘하는데 태도가 매우 중요한 만큼, SW교육에서 학습자가 SW와 자신과의 관계성을 깨닫는 것이 매우 중요하다. SW가 비전공자도 배울 수 있다는 자신감을 지각하고, 실생활의 문제를 해결하는데 유용한 도구가 될 수 있다고 인식하는 것이 SW학습에 대한 동기와 긍정적 태도를 형성하게 된다. 특히 SW역량은 한두 과목 이수로 달성하기 어렵다. 비전공자의 경우에도 여러 개의 과목을 체계적인 교육과정에 따라 학습하거나 관련 활동에 참여가 필요한데, 이러한 지속적인 학습행동은 SW학습에 대한 과제가치 인식과 흥미 없이 기대하기 힘들다. 교육과정이 심화되면

자연스럽게 내용의 난이도도 증가하지만 학습자가 어려움 속에서도 포기하지 않고 학습과 문제해결을 위한 집요한 노력을 발휘하려면 학습자가 지닌 정의적 요소가 큰 힘을 작용하게 된다. 반대로 SW학습에 대한 흥미와 동기를 잃어버리게 되면, 이를 회복하는 것도 쉽지 않으며 실제 비전공자들의 교육에서 동기 부족을 가장 어려운 점으로 꼽기도 한다[11]. 따라서 교육과정에 대한 평가 지표로 정의적 요소를 포함하여 관리하는 것이 장기적인 관점에서 SW교육과정 개발과 운영에 도움을 줄 수 있을 것이다. 교육과정 개발 시 학습자의 흥미와 동기 증진을 위한 고려가 필요한데, 학생들은 SW학습이 자신의 전공이나 생활에 어떻게 접목될 수 있는지 관련성을 파악할 수 있을 때 동기가 증가될 수 있다. 본 연구에서 교과는 학과와 교육과정 설계팀이 협력하여 운영하는 교과로, 학기마다 사전에 사회적 이슈와 학생의 관심사를 분석할 뿐 아니라 전공별로 전공의 특성을 고려한 대주제를 선정하고 있다. 이러한 노력은 학생들이 문제 상황에 공감하고 자연스럽게 프로젝트 해결에 몰입을 증진할 수 있다. 동료 학습자와 상호작용을 촉진하는 수업으로 설계하는 것도 학습자의 동기를 향상시키는데 도움이 될 수 있다. 교양강좌의 경우 학생들간 라포를 형성할 수 있도록 팀 빌딩을 할 수 있는 시간을 충분히 제공하고, 학점에 대한 경쟁보다는 학생들이 각자의 장점을 찾아가며 협력하여 결과물을 산출해내는 도전적 경험에 가치를 두고 격려하는 것이 비전공자들의 SW학습에서 긍정적인 경험과 인식을 강화할 수 있다. 요약하면, 학생들의 요구와 특징을 분석하고, 이를 반영하는 사례와 프로젝트 주제를 발굴하며, 교육과정에 대한 인지 및 정의적 요소를 포함한 평가를 토대로 교육과정을 지속적으로 개선하는 시스템적 교육과정 운영을 제안한다.

본 연구는 다음과 같은 한계를 지니고 이를 보완하는 후속 연구를 기대한다. 첫째, 하나의 대학에서 운영되는 교과목을 대상으로 자료를 수집하였으므로 연구결과의 일반화에 유의할 필요가 있다. 타대학의 비전공자 대상으로 한 후속 연구가 이루어진다면 연구결과에 대한 재확인과 일반화에 기여할 것이다. 둘째, 인문사회계열 학생에 초점을 두어 연구를 하였으나 타계열과의 비교는 이루어지지 않은 점이 제한점으로 연구결과를 유의해서 해석해야 한다. 셋째, 신입생을 대상으로 한 SW교육성과를 측정하였다. 다수 대학이 SW 관련하여 다수 학점을 취득하도록 하고 있는 상황이다. SW학습에 대한 과제가치 인식과 컴퓨팅 사고

력 수준에 따라 향후 수강에 영향을 미치는지, 교과목 수강 이력이나 시간이 지나면서 전공계열 간 차이를 보이는지 등의 후속 연구가 이루어진다면 대학의 SW 교육과정 개발과 운영에 기초자료를 제공할 것이다.

참고문헌

- [1] Park, J. (2021). Analysis of the effectiveness of computational thinking classes for improving convergent thinking skills. *The Journal of Korean Association of Computer Education*, 24(1), 71-80.
- [2] Lee, C. (2017). Development of real-life problem solving model(CT-RLPS Model) based on computational thinking for software education. *Journal of Korean Practical Arts Education*, 30(3), 33-57.
- [3] Pi, S. (2016). A study of coding education of non-computer majors for IT Convergence Education. *Journal of Digital Convergence*, 14(10), 1-8.
- [4] Kim, J. (2017). Computational thinking-oriented problem solving skills. *Korea Information Processing Society Review*, 24(2), 13-21.
- [5] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- [6] Hong, M., & Cho, J. (2021). A conceptual study on computational thinking for non-majors. *Journal of Convergence for information Technology*, 11(10), 151-158.
- [7] Chung, H. (2021). Analysis of algorithm thinking based on basic software education effectiveness for non-major students in SW college. *Korean Journal of General Education*, 15(2), 175-187.
- [8] Hong, S., Goo, E., Shin, S., Lee, T., & Seo, J. (2021). Development the measurement tool on the software educational effectiveness for non-major undergraduate students. *The Journal of Korean Association of Computer Education*, 24(1), 37-46.
- [9] Kim, S. (2015). Analysis of non-computer majors' difficulties in computational thinking education. *The Journal of Korean Association of Computer Education*, 18(3), 49-57.
- [10] Oh, M. (2017). Non-major students' perceptions of programming education using the Scratch programming language. *The Journal of Korean Association of Computer Education*, 20(1), 1-11.
- [11] Park, G., & Choi, Y. (2018). Exploratory study on the direction of software education for the non-major undergraduate students. *Journal of Education & Culture*, 24(4), 427-440.
- [12] Paik, S. H., & You, J. W. (2022). The interaction effect between gender and learning attitude on learning outcomes in AI education for elementary students. *The Journal of Korean Association of Computer Education*, 25(3), 49-60.
- [13] Lim, B. (2016). App Inventor 2 as a tool for enhancement of computational thinking. *Journal of The Korean Association of Information Education*, 20(5), 519-526.
- [14] Czerkawski, B., & Lyman, E. (2015). Exploring issues about computational thinking in higher education. *TechTrends*, 59(2), 57-65.
- [15] Ministry of Education. (2015). Curriculum of practical arts (technology and home economics)/information studies. https://ceri.knue.ac.kr/pds/2015_06_practical.pdf
- [16] Shin, C. (2021). The effects of computational thinking-based liberal education on problem solving ability. *Journal of the Korea Institute of Information and Communication Engineering*, 25(2), 246-251.
- [17] Park, Y., & Lee, M. (2020). A study on improving computational thinking education of university of by reflecting learner's perception and instructor's opinion. *Korean Journal of General Education*, 14(1), 167-191.
- [18] Papert, S. (2000). What's the big idea? Toward a pedagogy of idea power. *IBM Systems Journal*, 39(3-4), 720-729.
- [19] Flegg, J., Mallet, D., & Lupton, M. (2012). Students' perceptions of the relevance of mathematics in engineering. *International Journal of Mathematical Education in Science and Technology*, 43(6), 717-732.
- [20] Patton, E. W., Tissenbaum, M., & Harunani, F. (2019). MIT App Inventor: Objectives, design, and development. In S. -C. Kong and H. Abelson(Eds.), *Computational thinking education* (pp. 31-49). Springer.
- [21] Choi, H., & Kim, K. (2015). The effects of Scratch programming on preservice teachers: Assessment utilizing computational thinking and Bloom's taxonomy. *Journal of The Korean Association of information Education*, 19(2), 225-232.
- [22] Chung, Y., & You, J. W. (2020). Examining the effects of hackathon programs for first year students at arts and humanities colleges: Focusing on problem-solving competency and self-efficacy according to challenge orientation. *Korean Journal of General Education*, 14(6), 267-281.
- [23] Sung, J., S., Kim, S., & Kim, H. (2015). Analysis of art and humanity major learners' features in program-

- ming class. *The Journal of Korean Association of Computer Education*, 18(3), 25-35.
- [24] Lee, Y. J., Paik, S. H., Shin, J. H., Yoo, H. C. Jeong, I. K., Ahn, S. J., et al. (2014). *Research for introducing computational thinking into primary and secondary education*. KOFAC.
- [25] CSTA & ISTE. (2011). *Operational definition of computational thinking for K-12 education*. <http://www.iste.org/docs/pdfs/Operational-Definition-of-Computational-Thinking.pdf>
- [26] Lee, J. (2020). *Problem solving process-oriented computational thinking assessment*. Unpublished master's thesis, Jeju National University.
- [27] Lee, M., & Kim, S. (2019). Study on the development of a general-purpose computational thinking scale for programming education on problem solving. *The Journal of Korean Association of Computer Education*, 22(5), 67-77.
- [28] Lawanto, O., Santoso, H. B., Goodridge, W., & Lawanto, K. N. (2014). Task value, self-regulated learning, and performance in a web-intensive undergraduate engineering course: How are they related. *Journal of Online Learning and Teaching*, 10(1), 97-111.
- [29] Miller, R. B., & Brickman, S. A. (2004). A model of future oriented motivation and self-regulation. *Educational Psychology Review*, 16, 9-33.
- [30] Durak, H. Y., Yilmaz, F. G. K., & Yilmaz, R. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Education Technology*, 10(2), 173-197.
- [31] Yoo, I. (2014). Design a programming education plan for SW education using robot and mobile application development tool. *Journal of The Korean Association of information Education*, 18(4), 615-624.
- [32] Schiller, J. I., Turback, F., Abelson, H., Soler, J. L., McKinney, A., Okerlund, J., & Friedman, M. (2014). *Live programming of mobile apps in App Inventor*. Proceedings of the 2nd workshop on programming for Mobile & Touch (PROMOTO '14), Oregon, Association for Computing Machinery (ACM).
- [33] Kim, B. (2013). Computer programming education using App Inventor for android. *Journal of the Korea Institute Of Information and Communication Engineering*, 17(2), 467-472.
- [34] Jeon, S., & Lee, Y. (2015). The influence of learning App Inventor programming of LT collaborative learning based on children's motivation. *The Journal of Korean Association of Computer Education*, 18(2), 1-9.
- [35] Kang, E. H. (2019). *A study on the improvement of technology management system for personal information in the smart information society*. Unpublished master's thesis, Hanbat National University.
- [36] Deng, X. (2017). *Group collaboration with App Inventor*. Unpublished master's thesis, Massachusetts Institute of Technology.
- [37] Artino, A. R., & McCoach, D. B. (2008). Development and initial validation of the online learning value and self-efficacy scale. *Journal of Educational Computing Research*, 38(3), 279-303.
- [38] Chung, J. S., & Lim, K. Y. (2000). Effect analysis of factors related to the learner participation, achievement, and satisfaction in the web-based online discussion. *Journal of Educational Technology*, 18(2), 107-135.
- [39] The jamovi project (2021). *Jamovi*. (Version 1.6) [Computer Software]. Retrieved from <https://www.jamovi.org>
- [40] R Core Team (2020). *R: A language and environment for statistical computing* (Version 4.0) [Computer software]. Retrieved from <https://cran.r-project.org>
- [41] Cohen, J. (1992). A power primer. *Psychological Bulletin*, 112(1), 155-159.



유 지 원

1991년 이화여자대학교
전자계산학과(이학사)
1996년 University of Minnesota
교육공학전공(교육학석사)
2011년 이화여자대학교 교육공학과
(교육공학박사)

2014년 ~ 현재 가천대학교 교육대학원 AI융합교육전공 부교수
관심분야: 테크놀로지 기반 학습 설계 및 개발, AI융합교육,
SW교육, 학습성과 분석, 팀 기반 학습
E-Mail: uimagine@gachon.ac.kr