Check for updates

컴퓨터교육학회 논문지 2026년 제29권 제9호
https://doi.org/10.32431/kace.2026.29.9.011

초등학생 대상 프로그래밍 멘토링 교육이 대학생 멘토의 컴퓨팅 프로그래밍 자아효능감과 교수효능감에 미치는 효과

The Effects of Programming Mentoring for Elementary School Students on College Student Mentors' Programming Self-Efficacy and Teaching Self-Efficacy

홍수민† · 박성경†† · 박가은††† · 한도희†† · 김준서††† · 권혜성†† · 이은서†† · 김가형†††† · 권현지††††† · 임철일†††††††
Sumin Hong† · Sungkyung Park†† · Gaeun Park††† · Dohee Han††† · Junseo Kim††† · Hyeseong Kwon†† · Eunseo Lee†† · Gahyung Kim†††† · Cheolil Lim†††††††

요약

본 연구는 인공지능 교육에 대한 사회적 관심이 높아지는 가운데, 초등학생 프로그래밍 멘토링 교육이 대학생 멘토에게 미치는 영향과 그 효과를 탐색하고자 하였다. 먼저 선행연구 분석을 통해 프로그래밍 교육 전략과 디지털 격차 해소 전략을 포함한 멘토 대상 사전 워크숍을 설계하였다. 프로그램 참여 후 멘토의 프로그래밍 자아효능감은 모든 하위 요인에서 향상되었으며, 프로그래밍 태도의 경우 자신감 요인이 유의미한 변화를 나타냈다. 또한 멘토의 교수효능감 역시 향상되어, 멘토링 경험이 멘토의 교수 역량 개발에도 긍정적인 역할을 하였음을 확인하였다.

주제어 멘토링 프로그램, 프로그래밍 교육, 프로그래밍 자아효능감, 교수 효능감, 대학생 멘토

ABSTRACT

This study aimed to explore the effects of a programming mentoring program for elementary school students, led by university student mentors, amid growing social interest in artificial intelligence education. First, a pre-workshop for mentors was designed based on a review of prior literature, incorporating programming education strategies and digital divide mitigation strategies. Following program participation, mentors showed significant improvements in all sub-factors of programming self-efficacy, while in programming attitude, only the confidence factor demonstrated a significant change. In addition, mentors' teaching efficacy also improved significantly, confirming that the mentoring experience played a positive role in developing mentors' instructional competencies.

Keywords Mentoring Program, Programming Education, Programming Self-efficacy, Teaching Efficacy, College Student Mentor

†정회원 명지대학교 청소년지도학과 조교수
††정회원 서울대학교 대학원 교육학과 박사과정
†††정회원 서울대학교 대학원 교육학과 석사과정
††††정회원 서울대학교 학습과학연구소 선임연구원
†††††정회원 고려대학교 세종SW중심대학사업단 연구교수
††††††정회원 서울대학교 임철일 교수(교신저자)

논문투고 2026년 05월 16일
심사완료 2026년 07월 20일
게재확정 2026년 08월 31일
발행일자 2026년 09월 30일

1. 서론

디지털 전환 시대에 따라 인공지능 교육의 중요성이 전 교육 현장에서 증가하고 있다. 교육부는 2025년 11월 모두를 위한 인공지능(AI) 인재 양성 방안을 발표했으며, 초등학교부터 평생교육까지 전 생애에 걸친 인공지능(AI) 보편 교육을 강화한다고 강조하였다. 이때, 인공지능 교육을 위해서는 컴퓨팅 사고력(Computational Thinking)과 프로그래밍 능력이 요구된다. 인공지능의 원리를 이해하고, 이를 통해 문제를 해결하기 위해서는 데이터 처리, 알고리즘적 사고, 코딩 등 프로그래밍 소양이 선행되어야 하기 때문이다[1]. 이에 따라 2022 개정 교육과정에서는 초등학교부터 SW·AI 교육 시수를 확대하고, 프로그래밍을 포함한 정보 교육을 필수화하는 방향으로 전환되고 있다. 그러나 이러한 정책적 요구가 교육 현장에 실효성 있게 구현되기 위해서는 SW·AI 교육 및 프로그래밍을 효과적으로 지도할 수 있는 교수자의 역량 확보도 중요한 과제로 남아 있다.

실제 현장에는 전담 교원 부족 등의 어려움이 존재하며, 이에 대한 대안으로 방과후 수업 운영과 함께 대학생이 준전문가 멘토로 참여하는 멘토링 프로그램이 중요한 역할을 수행해 왔다. 멘토링 프로그램은 초등학생에게는 학습 경험을, 대학생에게는 교육봉사 및 실천적 교수 경험의 기회를 동시에 제공한다는 점에서 이점이 있다. 특히, 대학생 멘토는 학생과의 심리적 거리가 가깝게 친근한 교수 역할을 수행할 수 있어 학습 효과 측면에서도 긍정적으로 평가된다[2].

그러나 이와 관련한 정보 교육에서의 대학생-초등학생 멘토링 연구가 부족하며, 멘토링 프로그램에 대한 선행연구 역시 주로 멘티의 학습 성과에 초점을 두고 있어[3], 멘토의 변화에 주목한 연구는 상대적으로 적은 상황이다. AI·SW 시대 대학생의 프로그래밍 역량 함양이 고등교육의 핵심 과제로 부상한 현시점에서, 프로그래밍 교육 멘토링이 멘토로 참여하는 대학생의 프로그래밍 자아효능감 및 교수효능감에 미치는 영향을 규명하는 것은 학술적, 실천적 측면에서 중요하다.

따라서 본 연구에서는 멘토로 참여하는 대학생의 프로그래밍에 대한 변화에 주목하고자 한다. 'Learning by Teaching' 관점에 따르면, 가르치는 행위 자체가 교수자의 인지적·정의적 발달을 촉진하는 유의미한 학습 경험이 될 수 있다[4]. 특히 컴퓨팅·프로그래밍 분야에서는 직접 설계하고 실행하는 실습 중심의 교수 경험이 프로그래밍에 대한 자아효능감과 태도 변화에 긍정적인 영향을 미친다[5]. 나아가 멘토링 경험은 교수 효능감 형성의 요소로 작용할 수 있으며 예비교원 및 비전공 대학생의 교수 역량 개발 측면에서도 중요한 역할을 할 수 있다.

이에 본 연구는 초등학교를 대상으로 한 프로그래밍 멘토링 교육이 대학생 멘토에게 미치는 효과를 탐색하는 것을 목적으로 한다. 구체적으로, 프로그래밍 교육의 자아효능감 변화, 프로그래밍 교육에 대한 태도, 교수 효능감 변화를 확인하고자 한다.

한편, 멘토로 참여하는 대학생이 교수 경험의 효과를 충분히 발휘하기 위해서는 사전 준비가 중요하다. 체계적으로 설계된 멘토 사전 워크숍은 대학생이 프로그래밍 멘토 역할에 대한 준비도를 높이고, 멘토링 과정에서의 실천적 역량을 갖추도록 돕는 핵심 역할을 할 것이다. 따라서 멘토링의 효과를 높이기 위한 사전 워크숍을 선행연구를 기반으로 설계하고자 한다. 구체적인 연구 문제는 다음과 같다.

연구문제 1. 멘토링 프로그램을 위한 멘토 사전 워크숍은 어떻게 설계되었는가?

연구문제 2. 멘토링 프로그램에 참여한 후 멘토의 컴퓨팅 사고력의 자아효능감에 대한 유의미한 차이가 있었는가?

연구문제 3. 멘토링 프로그램에 참여한 후 멘토의 교수효능감에 대한 유의미한 차이가 있었는가?

2. 선행연구 분석

2.1 초등학생을 위한 프로그래밍 교육

국내 초등학교 프로그래밍 교육은 교육과정 개정을 통해 점진적으로 강화되어 왔다. 초등학교 프로그래밍 교육은 '도입기-단절기-강화기-체계화기'의 흐름을 거쳐 발전해왔으며, 2015 개정 교육과정에서 실과 교과를 통해 처음으로 필수화된 이후 2022 개정 교육과정에서는 디지털 소양이 기초 소양으로 명시되면서 정보 교육 시수가 34시간 이상으로 확대되고 데이터·인공지능 영역이 새롭게 추가되었다[6]. 특히 스크래치와 같은 블록형 프로그래밍 언어는 문법적 장벽을 낮추어 학생들이 컴퓨팅 사고력 함양에 집중할 수 있도록 돕는다[7].

초등 프로그래밍 교육은 학습자의 사고력 및 문제 해결 능력 발달에 긍정적 효과를 미친다. 나청수 외[8]에 따르면 SW 교육이 컴퓨팅 사고력의 하위 영역 중 수행과 개념 영역에서 효과가 있으며, 이철현[9]은 실생활 맥락을 활용한 프로그래밍 교육이 기존 시범·실습 방식보다 효과적임을 확인하였다. 한편 신승기[6]는 교육과정 이원화 편성으로 인한 학습 공백과 초등학교에 정보 교과가 별도로 편제되지 않은 구조적 한계로 인해 단위 학교 및 교사에 따른 프로그래밍 교육의 편차가 발생할 수 있음을 지적하였다. 이러한 맥락에서 대학생 멘토가 초등학생에게 프로그래밍 교육을 제공하는 멘토링 기반 접근은 학교 내 자원의 한계를 보완하는 대안적 교육 모델로서의 가능성을 지닌다.

2.2 프로그래밍 교육을 위한 멘토링

프로그래밍 교육에서 멘토링은 학습자의 인지적, 정의적 역량을 동시에 향상시키는 교수 전략으로 광범위하게 연구되어 왔다. 멘토링은 청소년, 학업, 직업 등 다양한 맥락에서 멘티의 행동적, 태도적, 동기적 성과 전반에 걸쳐 긍정적 효과와 연관되는 것으로 보고되었으며[10], 특히 컴퓨터과학 멘토링에서도 학습자의 프로그래밍 자아효능감과 CS에 대한 흥미 향상에 유의미한 기여를 하는 것으로 나타났다[2, 11].

또한, 프로그래밍 교육 멘토링은 멘티뿐 아니라 멘토 자신에게도 양방향적 성장의 기회를 제공할 수 있다. 대학 프로그래밍 멘토링 서비스에 관한 연구에서 멘토링이 초보 학습자의 프로그래밍 능력 향상에 기여할 뿐만 아니라, 멘토 자신의 자신감 증진과 의사소통 기술 향상이라는 부가적 효과를 수반하는 것으로 보고된 바 있다[12]. 이후 연구들에서도 멘토 역할 수행이 컴퓨터과학에 대한 자아효능감을 유의미하게 향상시키는 것으로 나타났다[11]. 이러한 양방향적 효과는 대학생이 멘토로 참여하는 프로그래밍 교육이 멘티인 초등학생의 학습 경험뿐 아니라, 멘토인 대학생의 교수 역량과 자아효능감 발달에도 의미 있는 교육적 맥락을 제공할 수 있음을 시사한다.

멘토링의 효과적 운영을 위해서는 멘토-멘티 간 지속적이고 구조화된 상호작용이 중요하다. 각 멘토를 소규모 학생 그룹에 고정 배치하여 세션 전체에 걸쳐 일관된 지원을 제공한 것이 멘티의 역량, 자율성, 관계성 형성에 핵심적으로 기여한 것으로 나타났으며[13], 연령과 경험이 가까운 멘토를 배치함으로써 밀접한 상호작용을 통해 멘티들이 대리적 경험을 통해 자아효능감을 형성할 수 있다는 점이 확인되었다[2]. 이는 멘토링 프로그램의 설계 시 단순한 멘토 배치를 넘어, 관계의 지속성과 구조적 안정성을 확보하는 것이 학습 효과를 높이는 데 필수적임을 보여준다.

또한, 프로그래밍 멘토링에서는 체계적 교수 전략이 요구된다. 비형식 멘토링 프로그램 교육의 운영 결과, 정해진 교수법이 부재하여 멘토들이 명확한 학습 이론을 적용하는 경우가 드물다는 한계가 지적되었다[14]. 비형식 프로그래밍 멘토링 교육 환경의 멘토들이 전통적 교실과는 상이한 방식으로 학습 환경을 구성하고 있으나, 교수학적 지식의 부족이 멘토링의 질적 편차를 야기할 수 있음이 확인되었다[15]. 이러한 연구 결과는 멘토링 프로그램의 효과를 향상시키기 위해서는 멘토에 대한 사전 훈련과 체계적인 교수 전략의 제공이 선행되어야 함을 강조한다.

2.3 프로그래밍 멘토링을 위한 사전학습의 중요성

프로그래밍 교육에서 멘토링이 효과적으로 운영되기 위해서는 멘토가 학습자의 수준을 이해하고 적절한 교수 전략을 적용할 수 있도록 사전 교육이 제공될 필요가 있다. 멘토링 프로그램에서 멘토 교육은 프로그램의 성공적인 운영과 학습 지원의 질을 결정하는 핵심 요소로 보고되어 왔으며, 체계적인 사전 훈련은 멘토가 학습자의 문제 해결 과정을 효과적으로 지원하고 학습 환경을 조성하는 데 중요한 역할을 한다[16]. 이러한 맥락에서 멘토 사전 교육의 방법 중 하나인 워크숍은 참여 중심의 학습 환경을 제공하는 형태로 활용되어 왔다. 워크숍은 참여자 간 상호작용과 활동 중심 학습을 통해 지식을 구성하도록 지원하며, 교수자 중심의 강의 방식보다 학습자의 적극적인 참여와 경험을 촉진하는 교육 방식으로 활용된다[17].

특히 프로그래밍 교육과 같은 실습 중심 학습에서는 학습자가 문제 해결 과정을 직접 수행하고 이를 반복적으로 수정하는 활동이 중요하며, 이러한 과정에서 학습을 지원하는 지도자의 역할이 중요한 것으로 드러난다[18]. 예를 들어, 스크래치를 활용한 프로그래밍 교육 연구에서는 블록형 프로그래밍 환경이 학습자가 프로그래밍의 문법적 부담 없이 창의적 문제 해결 활동에 참여하도록 지원하며, 이러한 학습 과정에서 상호작용과 협력적 활동이 학습 경험을 확장하는 데 중요한 역할을 한다는 점이 제시되었다[7]. 또한 멘토링 프로그램은 학습자의 인지적 성취뿐 아니라 학습 동기와 자아효능감 향상에도 긍정적인 영향을 미치는 것으로 보고되고 있다[10].

그러나 초등학생을 대상으로 한 프로그래밍 교육 멘토링에서 멘토를 위한 체계적인 사전 교육 프로그램의 설계와 운영, 효과를 다룬 연구는 아직 제한적인 상황이다. 대학생이 멘토로 활동하는 프로그래밍 교육 프로그램의 경우 멘토가 교육 경험이 부족한 경우가 많기 때문에, 효과적인 학습 지원을 위해서는 멘토가 학습자 수준에 맞는 설명 방식과 교수 전략을 이해할 수 있도록 구조화된 교육 과정이 필요하다[15]. 따라서 본 연구에서는 초등학생 대상 프로그래밍 교육 멘토링 프로그램을 운영하기에 앞서 대학생 멘토를 위한 워크숍을 설계하고 운영하고자 한다. 이를 통해 멘토가 프로그래밍 개념을 학습자의 수준에 맞게 설명하고 학습 활동을 지원할 수 있는 교수 역량을 함양하도록 지원하고, 이러한 경험이 멘토의 프로그래밍 교육에 대한 인식과 자아효능감에 어떠한 영향을 미치는지를 탐색하고자 한다.

3. 연구방법

3.1 연구 참여자

본 연구는 효과적인 멘토링 운영을 위해 워크숍이 어떻게 설계되고, 멘토링에 참여한 후 대학생의 컴퓨터 프로그래밍 자아효능감, 교수효능감에 대한 효과를 확인하는 것을 목적으로 한다. 이를 위해 서울 소재 S대학교 학생 30명을 연구 대상으로 하였다. 30명의 학생은 서울 소재 A 초등학교와 B 지역아동센터에서 멘토링 활동에 참여하였으며, 대학교 1학년부터 4학년까지 다양한 전공으로 구성되었다. 이 중 26명은 A 초등학교에, 나머지 4명은 B 지역아동센터에 배치되어 활동하였으며, 사전, 사후 설문에 모두 응답한 26명을 연구 대상으로 하였다.

3.2 연구 절차

본 연구의 절차는 다음과 같다. 멘토 30명을 대상으로 멘토링 시작 전 사전 설문을 시작하여 기초 자료를 수집하였다. 이후 멘토를 대상으로 선행연구 분석을 바탕으로 설계한 워크숍을 진행하였다.사전 워크숍을 통해 멘토링 내용과 절차를 표준화하고 멘토 간 실행 차이를 최소화한다. 멘토링은 2시간씩 총 10회 동안 운영되었으며, 각 회차는 대표 멘토의 전체 강의와 대학생 멘토의 소집단 멘토링으로 구성되었다. 먼저 대표 멘토가 해당 주차의 학습 목표와 주요 코

딩 개념을 설명하고, N사의 블록 기반 코딩 프로그램을 활용한 예제 활동을 시연하였다. 해당 프로그램은 블록 기반 코딩 교육 플랫폼으로, 게임의 아바타, 맵 등을 활용하여 학습자가 블록 코딩으로 자신만의 게임을 제작 및 공유할 수 있도록 설계된 게이미피케이션 기반 플랫폼이다. 전체 강의 이후 대학생 멘토는 약 5명 내외의 초등학생 멘티를 담당하여 활동지에 기반한 코딩 실습과 게임 제작 활동을 지원하였다. 이 과정에서 멘토는 멘티의 질문에 응답하고, 코딩 과정에서 발생하는 오류를 함께 점검하며, 멘티의 수준과 진행 속도에 따라 개별 피드백을 제공하였다. 예를 들어, 초반 회기에는 오브젝트와 아바타 제작, 기본 블록 구조, 시작블록과 신호블록 이해 등 기초 개념을 다루었고, 중반 회기에는 선택구조, 판단블록, 변수 개념을 활용한 문제 해결 활동을 진행하였다. 후반 회기에는 멘티가 자신만의 이야기 기반 게임을 제작하고, 오류를 수정하며, 최종 산출물을 발표하는 활동으로 확장되었다. 대학생 멘토는 매주 멘토링 활동에 앞서 해당 주차의 강의자료와 활동지를 사전에 검토하였으며, 멘토링 종료 후에는 활동 과정, 멘티의 참여 양상, 지도 과정에서의 어려움 및 개선점을 포함한 멘토 일지를 작성하였다. 마지막으로 멘토링 종료 후 사후 설문을 실시하여 변화를 측정하고 그 효과를 분석하였다.

3.3 연구 도구 및 분석 방법

본 연구에서는 멘토링의 효과를 확인하기 위해 Table 1과 같이 총 3가지 척도로 설문 문항을 구성하였다. 첫째, 컴퓨팅 프로그래밍 자아효능감은 Tsai et al. (2019)의 Computer Programming Self-Efficacy Scale을 기반으로 측정하였다[19]. 해당 척도는 논리적 사고, 협동, 알고리즘, 프로그램, 디버깅의 5개 하위영역 총 16문항으로 구성된다. 프로그래밍 태도는 Wiebe et al. (2003)의 척도를 기반으로 자신감, 성공 기대, 유용성, 동기의 4개 하위영역, 총 12문항으로 구성된다[20]. 교수 효능감은 Tschannen-Moran & Hoy(2001)의 척도를 대학생 멘토가 초등학생 멘티를 대상으로 프로그래밍 교육을 수행하는 본 연구의 멘토링 맥락에 맞게 수정, 보완하여 사용하였다[21]. 원척도의 하위영역인 학생참여, 교수전략, 학급관리의 구조는 유지하여, 이를 각각 멘티 참여 유도, 프로그래밍 교수전략, 멘토링 상황 관리로 재맥락화하였다. 해당 문항을 대학생 멘토-초등학생 멘티의 프로그래밍 교육 멘토링 맥락에 적합하도록 문항의 주어 및 어휘를 조정하였으며, 교육 분야 박사 학위 소지자 2명의 검토를 거쳤다. 본 연구의 모든 설문 문항은 6점 Likert 척도로 측정하였으며, 응답 범주는 1점 '전혀 그렇지 않다'에서 6점 '매우 그렇다'까지로 구성하였다. 본 연구에서 사용된 도구의 신뢰도를 확인하기 위해 Cronbach's α 계수를 산출한 결과, 사전 검사에서는 .981, 사후 검사에서는 .975로 나타나 매우 높은 신뢰도를 보였다.

Table 1. Components of survey

Category	Content	Reference
Computing Programming Self-efficacy	Logical Thinking	Tsai et al. (2019)
	Collaboration	
	Algorithm	
	Programming	
	Debugging	
Programming Attitude	Confidence	Wiebe et al.(2003)
	Success Expectation	
	Usefulness	
	Motivation	
Teaching efficacy		Tschannen-Moran& Hoy (2001)

멘토링 프로그램 참여 전·후의 변화를 검증하기 위해 대응표본 t 검정(paired samples t-test)을 실시하였다. 대응표본 t 검정은 동일한 집단을 대상으로 사전·사후의 평균 차이가 통계적으로 유의한지를 검증하는 방법으로, 본 연구와 같이 동일한 멘토 집단의 사전·사후 변화를 비교하는 데 적합한 분석 방법으로 유의수준은 .05로 설정하였다.

4. 연구결과

4.1 멘토 사전 워크숍

멘토링 교육 프로그램을 실행하기 전, 멘토링 프로그램을 효과적으로 운영하기 위하여 사전 워크숍을 설계 및 실행하였다. 멘토링 프로그램을 효과적으로 시행하기 위해서는 본 프로그램의 목적, 코딩 교육의 목표, 코딩 교육 설계 방법, 수업 사례 전달을 비롯하여 코딩 도구에 대한 충분한 이해, 프로그래밍 교육 전략, 멘토링 전략 등이 필요하다. 이에 블록 코딩 기반 교수학습 전략과 디지털 격차 멘토링 전략에 대하여 선행 문헌을 분석하였다. 첫째, 블록 코딩 교육 전략으로는 블록 코딩 기반 문제 해결 원리[22], 전략적 스캐폴딩 및 초인지 스캐폴딩[23], 컴퓨팅 사고력 함양을 위한 UMC 모형[24], 코드 작성 수준 측정 방법[25, 26]등에 대한 교육 전략을 확인하였다. 둘째, 교육 과정 중 고려하게 될 디지털 격차 해소 전략으로 단계별 난이도 조절, 소그룹 조력자 구성, 자유 탐색 시간 제공[27, 28], 디지털 멘토링 8원칙 등을 확인하였다. 문헌 검토를 통해 도출된 전략은 Table 2와 같다.

도출된 프로그래밍 멘토링을 위한 사전 워크숍 설계 전략과 실현 가능성을 검토하기 위해 교육공학 분야 박사 2인과 교육 분야 박사 1인 전문가를 대상으로 전문가 검토를 실시하였다. 전문가들은 각 전략의 맥락 적합성 및 멘토링 목표와의 부합 여부를 검토한 후 수정 의견을 제시하였다.

전문가 검토 결과 네 가지의 피드백이 제시되었으며, 이를 워크숍 설계에 반영하여 수정하였다. 첫째, 전략의 구조적 분류를 통해 프로그래밍의 순차적인 학습 단계에 따라 제시하여 멘토가 교육의 흐름을 따라 전략을 수월하게 이해하

고 적용할 수 있도록 재구조화하였다. 둘째, 팀 기반 활동 설계가 제시되었다. 프로그래밍 교육과정은 팀 활동으로 이루어지기에, 팀 활동의 효과를 높이기 위한 활동이 포함되도록 제안되었다. 셋째, 도구 탐색 설계의 적절성 확인으로, 자유 탐색 이후 간단한 미션 수행으로 탐색적 학습을 촉진하는 전략이 제안되었다. 넷째, 성찰 활동 구조화로, 참여자가 워크숍 마무리 단계에서 개별적으로 성찰하고 준비로 이어지도록 제안되었다.

문헌 분석과 전문가 검토 결과 및 피드백을 반영하여 개발된 최종 멘토 사전 워크숍 프로그램은 다음과 같다. 총 90분으로 구성되며, 프로그래밍에 사용하는 도구 이해, 실습 중심의 세션과 멘토링 전략 및 성찰 중심의 세션으로 운영된다. 약 30명의 참여자는 4-5명의 6팀으로 구성되어 전 과정을 팀으로 진행한다. 최종 프로그램의 세부 구성은 Table 3과 같다.

멘토링 프로그램에 참여하는 멘토 30명을 대상으로 2차례 진행하였다.

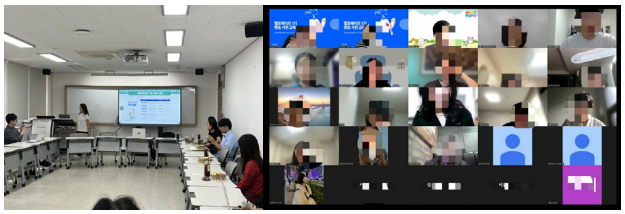

Figure 1. Pre-workshop for mentor

Table 2. Literature Review of Design Strategies

Domain	Strategy	Key Content	Reference
Block Coding & Mentoring Strategies	Principles of Block Coding-Based Problem Solving	Tailoring activities to learners' coding levels; providing real-life contextual tasks; structuring collaborative learning	Lim et al. (2023)
	Principles of Learner Support	Encouraging use of diverse coding blocks; providing sufficient time for code design and storyboard development	Lim et al. (2023)
	Strategic Scaffolding	Direct implementation of program examples to acquire diverse strategies; providing guiding questions and expert advice for task solving; meta-cognitive scaffolding through reflective thinking	Kim & Jeong (2011)
	Computational Thinking (UMC Model)	Sequenced lesson design: Use → Modify → Create	Lytle et al.(2019)
	Assessment of Code Proficiency	Evaluating computational thinking attainment via Dr. Scratch; providing rubric/matrix-based feedback	Kim (2015); Son (2016)
Digital Divide Mitigation Strategies	Difficulty Adjustment	Dividing tasks on the same topic into easy – basic – extended levels to promote self-directed learning and prevent frustration	Ahn (2023)
	Small-Group Facilitator Structure	Forming groups of 2 – 3; assigning digital facilitator roles to foster mutual cooperation and peer support	Ahn (2023)
	Free Exploration & Pre-instruction	Providing free exploration time after tool introduction; pre-briefing on the benefits of digital device use	Kim & Ku (2021)
	8 Principles of Digital Mentoring	Applying 8 principles: mentoring style, motivation, safe space creation, interest-driven learning, goal setting, and more	Digital Mentoring Handbook

Table 3. Workshop Program Overview

Session	Activity	Content	Theoretical Basis
Session 1	Orientation & Ice breaking	Sharing mentoring goals; exchanging expectations and personal strengths	Team-Based Learning (TBL)
	Tool Exploration & Experience	Introduction to key features followed by free exploration time	Free Exploration; Scaffolding (Kim et al., 2011)
	Mission-Based Tool Practice	Stepwise programming practice: easy → intermediate → advanced (UMC model applied)	UMC (Lytle et al., 2019); Ahn (2023)
Session 2	Mentoring Strategy Practice & Simulation	Digital divide learner support strategies; cross-team comparative role-playing	Strategic & Metacognitive Scaffolding (Kim et al., 2011); Ahn (2023)
	Wrap-up & Q&A	Sharing reflections; announcing upcoming mentoring schedule; Q&A	–

4.2. 멘토의 프로그래밍 자아효능감 및 태도

대학생 멘토가 멘토링 참여 이후 멘토들의 프로그래밍 자아효능감 및 태도의 변화를 확인하하였다. 분석 결과, 사전 평균(M = 4.75, SD = 1.07)보다 사후 평균(M = 5.39, SD = 0.60)이 높게 나타났으며, 이러한 차이는 통계적으로 유의하였다($t(25) = -4.496$, $p < .001$). 이는 멘토링 프로그램이 멘토의 컴퓨팅 사고력 자아효능감을 향상시키는 데 긍정적인 영향을 미쳤음을 보여준다.

멘토링 프로그램이 대학생 멘토의 프로그래밍 태도에 미치는 변화를 분석하기 위해 대응표본 t-검정을 실시하였다. 분석 결과, 사전 평균(M = 4.92, SD = 0.75)보다 사후 평균(M = 5.18, SD = 0.64)이 높게 나타났으며, 이러한 차이는 통계적으로 유의하였다($t(25) = -2.392$, $p = .024$). 이는 멘토링 프로그램이 멘토의 프로그래밍 태도를 향상시키는 데

긍정적인 영향을 미쳤음을 보여준다.

멘토의 컴퓨팅 사고력 자아효능감 하위요인에 미치는 변화를 분석하기 위해 5개 하위요인별로 대응표본 t-검정을 실시하였다. 분석 결과, 컴퓨팅 사고력 자아효능감의 5개 하위요인 모두에서 사전 대비 사후 점수가 통계적으로 유의하게 향상된 것으로 나타났다. 특히 협동($p < .001$, $d = 0.940$)과 알고리즘($p < .001$, $d = 0.836$) 영역에서 큰 효과 크기가 확인되었으며, 이는 멘토링 프로그램이 멘토들의 협력적 프로그래밍 경험과 절차적 사고 능력을 향상시키는 데 실질적인 효과가 있었음을 시사한다.

Table 4. Paired t-test result of CT self-efficacy subfactors (N=26)

	Subfactor	Group	M	SD	t	*p*
Computational thinking self-efficacy scale	Logical Thinking	Pre	5.27	0.83	-2.573	.016*
		Post	5.54	0.58		
	Cooperation	Pre	4.51	1.08	-4.791	.000***
		Post	5.29	0.82		
	Algorithm	Pre	4.26	1.35	-4.261	.000***
		Post	5.15	0.93		
	Programming	Pre	5.04	1.33	-2.637	.014*
		Post	5.63	0.63		
	Debugging	Pre	4.51	1.41	-3.195	.004**
		Post	5.31	0.59		

*$p < .05$, **$p < .01$, ***$p < .001$

멘토링 프로그램이 멘토의 프로그래밍 태도 하위요인에 미치는 변화를 분석하기 위해 4개 하위요인별로 대응표본 t-검정을 실시하였다[20]. 분석 결과, 프로그래밍 태도의 4개 하위요인 중 자신감 영역만이 통계적으로 유의한 향상을 보였으며($p = .003$, $d = 0.654$), 성공 기대, 유용성, 동기 영역에서는 유의한 변화가 나타나지 않았다. 이는 성공 기대, 유용성, 동기 영역의 사전 점수가 Likert 6점 척도 기준 이미 4.97~5.01점으로 높은 수준이었기 때문에 천장효과(ceiling effect)가 작용한 것으로 해석된다. 반면, 자신감 영역은 사전 점수($M = 4.71$)가 상대적으로 낮아 멘토링 프로그램을 통한 실질적인 프로그래밍 경험이 자신감 향상에 유의한 영향을 미친 것으로 판단된다.

Table 5. Paired t-test result of programming attitude sub-factors(N=26)

Content	Sub-factor	Group	M	SD	t	p
Programming Attitude	Confidence	Pre	4.71	1.18	-3.335	.003**
		Post	5.28	0.68		
	Attitude toward success	Pre	5.01	0.95	-0.799	.432
		Post	5.13	0.75		
	Usefulness	Pre	4.97	0.68	-0.411	.684
		Post	5.05	0.81		
	Motivation	Pre	4.97	0.84	-1.912	.068
		Post	5.24	0.75		

*$p < .05$, **$p < .01$, ***$p < .001$

4.3. 멘토의 교수효능감

멘토링 프로그램이 멘토의 교수효능감에 미치는 변화를 분석하기 위해 대응표본 t-검정을 실시하였다. 분석 결과 사전 평균($M = 4.82$, $SD = 0.83$)보다 사후 평균($M = 5.26$, $SD = 0.62$)이 높게 나타났으며 이러한 차이는 통계적으로 유의하게 나타났다($t(25) = -3.169$, $p = .004$). 이는 멘토링 프로그램이 멘토의 교수효능감을 향상시키는 데 긍정적인 영향을 미쳤음을 보여준다.

Table 6. Paired t-test result of Teaching efficacy (N=26)

	Group	M	SD	t	p
Teaching efficacy	Pre	4.82	0.83	-3.169	.004**
	Post	5.26	0.62		

이러한 변화를 보다 구체적으로 이해하기 위해 실시한 설문조사에 대한 개방형 응답을 질적으로 분석하였다. 질적 자료 분석은 교수효능감을 학생 참여(Student Engagement), 교수 전략(Instructional Strategies), 수업 관리(Classroom Management)의 세 요인으로 설명한 선행연구를 분석 틀로 활용하였다[21].

개방형 응답 분석 결과 멘토들의 자아효능감 경험은 교수 전략, 학생 참여 효능감 측면에서 두드러지게 나타났다. 먼저, 멘티의 학습 성취와 성장 경험을 확인하는 과정에서 멘토의 효능감이 강화되는 모습이 나타났다. 응답 중에서“코딩에 대해 잘 몰랐던 아이들도 마지막 회차에서 자신만의 게임을 완성해가는 모습”과“처음에는 코딩에 흥미를 갖지 못한 아이들이 활동을 통해 흥미를 갖고 나중에는 스스로 코딩하는 모습”에서 많은 보람과 의미를 느끼는 것을 통해 멘티의 학습 성취 경험이 멘토의 지도 능력에 대한 자신감을 형성하는 요인으로 작용했음을 확인할 수 있었다. 또한“학생이 만들고 싶은 프로그램을 힌트를 줘서 조건문 블록을 활용하면 좋을 것 같다고 했을 때 그 친구가 고민하다가 깨닫고 행복해하던 순간”과 같은 응답에서 멘토의 지도 개입이 학생의 문제 해결로 이어지는 경험이 멘토의 교수효능감 형성

에 긍정적으로 작용했음을 확인할 수 있었다.

마지막으로 멘티의 긍정적 반응과 상호작용 경험 역시 멘토의 교수효능감 형성과 관련된 것으로 나타났다. 멘토들은 멘티들이 수업을 즐기거나 감사의 표현을 하는 경험을 통해 자신의 지도 활동이 의미 있었다는 확신을 갖게 되었다. 예를 들어 “아이들이 재밌다고 해줄 때가 제일 보람이 있었습니다”, “마지막 수업 이후 멘티들이 수업을 더 이어서 하고 싶다고 말해주었을 때 보람을 느꼈다”, “멘티 학생들이 헬로 메이플 수업을 열심히 해줘서 고맙다고 인사했을 때”와 같은 응답이 나타났다. 이러한 경험은 멘토가 자신의 지도 활동이 학습자에게 긍정적인 영향을 미쳤다고 인식하게 하여 자아효능감을 강화하는 요인으로 작용한 것으로 해석된다.

5. 논의 및 결론

본 연구에서는 초등학교를 대상으로 한 멘토링 프로그램이 대학생 멘토에게 미치는 효과를 탐색하기 위해서, 멘토 대상 사전 워크숍을 설계하고, 멘토의 변화를 종합적으로 확인하였다. 연구에서 도출된 주요 결과를 바탕으로 논의한 내용은 다음과 같다.

첫째, 멘토링 프로그램 참여 후 멘토의 프로그래밍 자아효능감은 모든 하위 요인에서 유의미하게 향상된 반면, 프로그래밍 태도의 경우 자신감 요인이 통계적으로 유의한 변화를 나타냈다. 우선, 프로그래밍 자아효능감의 전반적인 향상은 멘토링이라는 교수 경험이 멘토의 역량 인식에 긍정적으로 작용한다는 것을 확인할 수 있었다. Clarke-Midura와 동료들(2016)은 멘토가 멘토링 이후 자아효능감이 유의미하게 향상되었음을 확인하였으며[11], D'Souza와 동료들(2008)은 멘토링이 학습자의 프로그래밍 능력 향상에 영향을 끼칠 뿐만 아니라, 멘토의 자신감 증진과 의사소통 기술 향상의 효과를 확인하였다[12]. 본 연구에서도 프로그래밍 교육 멘토링이 멘토의 프로그래밍에 대한 자아효능감과, 프로그래밍의 태도 특히, 자신감에 유의미한 효과가 나타난 것을 확인할 수 있었다.

이러한 결과는 프로그래밍 멘토링의 교육적 가치를 멘티 중심에서 멘토-멘티의 상호 성장 모델로 재해석할 필요성을 시사한다. 본 연구의 결과와 선행연구[11, 12]는 멘토 또한 멘토링 경험을 통해 자아효능감과 정의적 태도가 유의미하게 성장하는 학습 주체임을 보여주며, 이는 ‘가르치며 배운다(Learning by Teaching)’을 뒷받침한다[29]. 따라서 프로그래밍 멘토링은 초등 멘티의 SW 교육 격차 해소와 대학생 멘토의 역량 강화의 가능성을 보여준다.

둘째, 멘토링 참여 후 프로그래밍 자아효능감 뿐만아니라 멘토의 교수효능감 또한 유의미하게 향상된 것으로 나타났다. 이는 멘토링이 멘티의 학습 지원에 그치지 않고, 멘토의 교수 역량 개발에도 긍정적인 영향을 미침을 보여주는 결과라고 할 수 있다. 대학생 멘토들은 초등학생 대상 프로그래밍 수업을 직접 계획하고 실행하는 과정에서, 학습자의 이해 수준에 맞게 설명을 조정하고 적절한 피드백을 제공하며 학습동기를 유지시키기 위한 다양한 교수 전략을 실제 적용하는 경험을 하였다. 이러한 결과는 앞서 언급한 ‘가르치며 배운다’가 프로그래밍 멘토링 맥락에서 구체화된 사례로 해석할 수 있다.

이러한 실천적 교수 경험은 이론적 지식만으로 형성되기 어려운 교수 역량을 자연스럽게 촉진하는 장으로 역할하였으며, 이러한 경험이 교수효능감의 향상으로 작용한 것으로 해석된다. 이와 유사하게, 박명신과 임선희(2013)은 학습멘토링 프로그램에 참여한 학생 멘토들이 학점 이수 외에도 학습 경험과 리더십, 교수활동 경험에 실질적으로 도움을 받았다고 인식하였으며, 교사로서의 목적의식이 더욱 뚜렷해진 것으로 나타났다[30]. 또한 이인홍과 동학림(2023)은 멘토링 연구의 초점을 멘티에서 멘토로 이동시켜야 할 필요성을 강조하였다[31]. 이러한 맥락에서 본 연구는 프로그래밍 교육 분야에서 대학생 멘토를 대상으로 멘토링 참여가 교수효능감 향상으로 이어질 수 있음을 실증적으로 확인하였다는 점에서 의의가 있다.

한편, 본 연구는 다음과 같은 한계를 지닌다. 첫째, 멘토의 프로그래밍 자아효능감, 프로그래밍 태도, 교수효능감을 자기보고식 설문으로 측정하였기 때문에, 실제 프로그래밍 능력 향상이나 문제해결역량을 직접적으로 확인하지 못하였다. 향후 연구에서는 멘토를 대상으로 한 프로그래밍 평가, 문제해결역량 검사 등을 병행하여 보다 다각적으로 검토할 필요가 있다. 둘째, 본 연구는 멘토인 대학생의 변화에 초점을 맞추었기 때문에, 멘토링 프로그램의 직접적인 학습자인 멘티의 변화는 확인하지 않았다. 추후 연구에서는 멘토와 멘티를 동시에 추적하는 종단적 설계를 통해 멘토링 프로그램이 양측에 미치는 상호적 효과를 종합적으로 규명할 필요가 있다.

셋째, 본 연구는 서울 소재 단일 대학교의 멘토를 대상으로 한 단일집단 사전-사후 설계를 채택하였으며, 연구 결과를 다양한 대학생 멘토에게 일반화하는 데에는 한계가 있다. 향후 연구에서는 다양한 기관의 학생을 대상으로 하고, 통제집단을 포함한 준실험 연구가 수행될 필요가 있다. 넷째, 본 연구에서는 모든 대학생 멘토가 사전 워크숍에 참여한 후 멘토링 활동을 수행하였기 때문에, 사전 워크숍과 멘토링 활동의 효과를 분리하여 해석하는 데 한계가 있다. 따라서 본 연구 결과는 사전 워크숍과 멘토링 활동이 결합된 프로그램 효과로 해석할 필요가 있으며, 향후 연구에서는 비교집단 설정이나 단계적 측정을 통해 각 요소의 효과를 보다 정교하게 검증할 필요가 있다.

참고문헌

[1] Shamir, G., & Levin, I. (2026). Cultivating machine learning-related computational thinking in elementary school. *Thinking Skills and Creativity, 61*, 102177. https://doi.org/10.1016/j.tsc.2026.102177

[2] Zaniewski, A., & Reinholz, D. (2016). Increasing STEM success: A near-peer mentoring program in the physical sciences. *International Journal of STEM Education, 3*(1), 14. https://doi.org/10.1186/s40594-016-0043-2

[3] Sun, C., & Clarke-Midura, J. (2022). Testing the efficacy of a near-peer mentoring model for recruiting youth into computer science. Mentoring & Tutoring: Partnership in Learning, 30(2), 184–201. https://doi.org/10.1080/13611267.2022.2057101

[4] Hudson, P. (2013). Mentoring as professional development: 'Growth for both' mentor and mentee. *Professional Development in Education, 39*(5), 771–783. https://doi.org/10.1080/19415257.2012.749415

[5] Kim, S., & Lee, Y. (2019). The changes of self-efficacy beliefs of pre-service teachers for technology integration through programming-based TPACK educational program. *Journal of the Korea Society of Computer and Information, 24*(4), 185–193. https://doi.org/10.9708/jksci.2019.24.04.185

[6] Shin, S. (2025). Investigating learning gaps in elementary informatics education: Implications of the 2022 revised curriculum implementation. *Journal of Creative Information Culture, 11*(4), 469–483. https://doi.org/10.32823/jcic.11.4.202511.469

[7] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

[8] Na, C., Joo, H., Lee, J., & Kim, D. (2018). Inducing computational thinking in Korean SW education: Synthesizing standardized mean changes through meta-analysis. *Journal of Educational Technology, 34*(3), 775–815. https://doi.org/10.17232/KSET.34.3.775

[9] Lee, C. (2017). The effect of computational thinking-based real-life problem solving learning on improving elementary school students' computational thinking. *Journal of Korean Practical Arts Education, 23*(4), 91–107. https://doi.org/10.29113/skpaer.2017.23.4.091

[10] Eby, L., Allen, T., Evans, S., Ng, T., & DuBois, D. (2008). Does mentoring matter? A multidisciplinary meta-analysis comparing mentored and non-mentored individuals. *Journal of Vocational Behavior, 72*(2), 254–267. https://doi.org/10.1016/j.jvb.2007.04.005

[11] Clarke-Midura, J., Allan, V., & Close, K. (2016). Investigating the role of being a mentor as a way of increasing interest in CS. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, Memphis, TN, USA, 297–302. https://doi.org/10.1145/2839509.2844581

[12] D'Souza, D., Hamilton, M., Harland, J., Muir, P., Thevathayan, C., & Walker, C. (2008). Transforming learning of programming: A mentoring project. *Proceedings of the Tenth Australasian Computing Education Conference*, Wollongong, NSW, Australia, 78, 75–84.

[13] McLaughlin, G., Ahmadi, S., Novak, E., & Borgerding, L. (2025). Mentoring in K-12 computer science classrooms: Exploring mentors' and mentees' experiences. *ACM Transactions on Computing Education, 26*(1), 1–20. https://doi.org/10.1145/3772073

[14] Alsheaibi, A., Strong, G., & Millwood, R. (2018). The need for a learning model in CoderDojo mentoring practice. *Proceedings of the 13th Workshop in Primary and Secondary Computing Education, Potsdam*, Germany, 1–2. https://doi.org/10.1145/3265757.3265785

[15] Alsheaibi, A., Huggard, M., & Strong, G. (2020). Teaching within the CoderDojo movement: An exploration of mentors' teaching practices. *Proceedings of the 2020 IEEE Frontiers in Education Conference*, Uppsala, Sweden, 1–5. https://doi.org/10.1109/FIE44824.2020.9273998

[16] Le, H., Sok, S., & Heng, K. (2024). The benefits of peer mentoring in higher education: Findings from a systematic review. *Journal of Learning Development in Higher Education, 31*. https://doi.org/10.47408/jldhe.vi31.1159

[17] Olmstead, A., & Turpen, C. (2016). Assessing the interactivity and prescriptiveness of faculty professional development workshops: The real-time professional development observation tool. *Physical Review Physics Education Research, 12*(2), 020136. https://doi.org/10.1103/PhysRevPhysEducRes.12.020136

[18] Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

[19] Tsai, M., Wang, C., & Hsu, P. (2019). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research, 56*(8), 1345–1360. https://doi.org/10.1177/0735633117746747

[20] Wiebe, E., Williams, L., Yang, K., & Miller, C. (2003). Computer science attitude survey (Technical Report No. TR-2003-01). North Carolina State University, Department of Computer Science.

[21] Tschannen-Moran, M., & Hoy, A. (2001). Teacher efficacy: Capturing an elusive construct. *Teaching and Teacher Education, 17*(7), 783–805. https://doi.org/10.1016/S0742-051X(01)00036-1

[22] Lim, C., Lee, D., Jung, H., & Lee, U. (2023). Using block coding to develop the design principles and a model for an integrated Korean language class. *Journal of Educational Technology, 39*(2), 345–377. https://doi.org/10.17232/KSET.39.2.345

[23] Kim, S., & Jeong, I. (2011). The scratch programming learning attitude effects of scaffolding based learning strategy. Journal of the Korean Association of Information Education, 15(1), 39–49.

[24] Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J.,

Milliken, A., & Barnes, T. (2019). Use, modify, create: Comparing computational thinking lesson progressions for STEM classes. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, Aberdeen, Scotland, UK, 395–401. https://doi.org/10.1145/3304221.3319755

[25] Kim, S. (2015). Development of scratch code analysis system for assessment about concepts of computational thinking. The Journal of Korean Association of Computer Education, 18(6), 13–22.

[26] Sohn, W. (2016). A method for measuring and evaluating for block-based programming code. *Journal of the Korean Association of Information Education, 20*(3), 293–302.

[27] hn, J. (2023). Digital divide among the digital natives: The exploration of influencing factors and suggestions. *Journal of Speech, Media and Communication Research, 22*(2), 37–68.

[28] Kim, S., & Koo, H. (2021). Is the youth generation free from the digital divide? A study on the utilization of mobile information of the 2030 youth generation in Korea. *The Journal of the Korea Contents Association, 21*(1), 39–54. https://doi.org/10.5392/JKCA.2021.21.01.039

[29] Fiorella, L., & Mayer, R. (2013). The relative benefits of learning by teaching and teaching expectancy. *Contemporary Educational Psychology, 38*(4), 281–288. https://doi.org/10.1016/j.cedpsych.2013.06.001

[30] Park, M., & Lim, S. (2013). A case study on the effects and improvement of learning mentoring program for middle school students targeted education welfare. *Journal of Adolescent Welfare, 15*(2), 197–225.

[31] Lee, I., & Dong, H. (2023). The effect of mentoring on the mentor's job satisfaction: Mediating effects of personal learning and self-efficacy. Asia-Pacific *Journal of Business Venturing and Entrepreneurship, 18*(3), 157–172. https://doi.org/10.16972/apjbve.18.3.202306.157

홍수민

· 2026년 서울대학교 교육학과 교육공학전공(교육학박사)
· 2021년 서울대학교 교육학과 교육공학전공(교육학석사)
· 2025년 4월 ~2026년 8월 서울대학교 학습과학연구소 선임연구원
· 2026년 9월~현재 명지대학교 조교수

관심분야 : 교수설계, 인공지능 융합 교육, 몰입형 수업, 교사교육
shong@mju.ac.kr

박성경

· 2024년 서울대학교 교육학과 교육공학전공(교육학석사)
· 2021년 한동대학교 ICT융합학과 (공학학사)
· 2025년~현재 서울대학교 대학원 교육학과 박사과정

관심분야 : 교수설계, 교사교육
shuapark@snu.ac.kr

박가은

· 2025년 중앙대학교 사범대학 교육학과(교육학사)
· 2025년~현재 서울대학교 대학원 교육학과 석사과정

관심분야 : 교수체제설계, AI 기반 맞춤형 교육, 에듀테크
ga.eun@snu.ac.kr

한도희

· 2021년 공주대학교 컴퓨터교육과(교육학사)
· 2025년~현재 서울대학교 대학원 교육학과 석사과정
· 2021년~현재 인천 석정중학교 교사

관심분야 : AI융합교육, 생성형 AI 기반 수업 설계
ehgml910@snu.ac.kr

김준서

· 2025년 서울대학교 사범대학 교육학과(문학사)
· 2025년 서울대학교 공과대학 컴퓨터공학부(공학사)
· 2025년~현재 서울대학교 대학원 교육학과 석사과정

관심분야 : 교수체제설계, AI 융합 교육, 적응형 학습
jskim1124@snu.ac.kr

권혜성

· 2019년 서울대학교 교육학과 교육공학전공(교육학석사)
· 2012년 청주교육대학교 초등교육과(교육학사)
· 2025년~현재 서울대학교 대학원 교육학과 박사과정
· 2012년~현재 과천초등학교 교사

관심분야 : 교수체제설계, AI융합교육, AIDT 설계 및 활용
comet@snu.ac.kr

이은서

· 2022년 서울대학교 교육학과 교육공학전공(교육학석사)
· 2014년 대구교육대학교 초등교육과(교육학사)
· 2024년~현재 서울대학교 대학원 교육학과 박사과정
· 2012년~현재 서울성내초등학교 교사

관심분야 : 인공지능기반 교육 , Learning Analytics, Adaptive Learning, ISD
eunseolee@snu.ac.kr

김가형

· 2015년 중앙대학교 사범대학 영어교육학과(문학사)
· 2019년 서울대학교 대학원 글로멀교육협력(교육학석사)
· 2024년 서울대학교 대학원 글로벌교육협력(교육학박사
· 2024년~현재 서울대학교 학습과학연구소 선임연구원

관심분야 : 교육적 대안, 국제교육, AI·디지털 교육, 교사교육
ghkim91@snu.ac.kr

권현지

· 2024년 동국대학교 교육학과 인적자원개발 전공(교육학박사)
· 2025년~현재 고려대학교 세종SW중심대학사업단 연구교수

관심분야 : 진로교육, 인간-AI협업, AI 교육, 학습분석학
researchg@korea.ac.kr

임철일

· 1986년 서울대학교 사범대학 교육학과(학사)
· 1988년 서울대학교 사범대학 교육학과(교육학석사)
· 1994년 Indiana University Dept. of Instructional Systems Technology(교육학박사)
· 2004년~현재 서울대학교 교육학과 교수

관심분야 : 교수설계, 교수체제설계, 인공지능 융합 교육
chlim@snu.ac.kr