

컴퓨터교육학회 논문지 2026년 제29권 제8호
https://doi.org/10.32431/kace.2026.29.8.001



고급 프로그래밍 수업에서 생성형 AI 활용에 대한 학생 인식과 프로젝트 참여 사례 연구

Student Perceptions of Generative AI Use and Project Participation in an Advanced Programming Course: A Case Study

이은정[†]
Eunjung Lee[†]

요약

본 연구는 생성형 AI 활용이 공식화된 대학 프로그래밍 수업에서 학습자 인식과 협업 양상을 실증 분석하였다. 설문 결과, 학생들은 프로젝트 수행 시 생성형 AI 효능감을 높게 평가했으나, 전통적 평가 상황에서는 생성형 AI 의존성과 공정성 우려를 나타냈다. 한편 2년간의 팀 프로젝트 데이터 분석 결과, 생성형 AI 도입 후 팀별 기여도 분산은 11.4% 감소했으며, 무임승차 구간 학생들이 중간 참여층으로 이동하는 협업의 상향평준화가 확인되었다. 이를 바탕으로 본 연구는 생성형 AI 시대 프로그래밍 교육을 위한 기초 개념 학습과 프로젝트의 구성 방안, PBL의 다면 평가 방안, 학습자의 자기규율 역량 강화의 필요성을 고찰하였다. 다만, 특정 교과목의 소규모 사례 연구로서 분석 결과의 일반화에는 한계가 있는 바, 향후 다양한 학습자 집단과 객관적 데이터를 활용한 후속 연구를 통해 분석 모델의 정교화가 필요하다.

주제어 고급프로그래밍 교육, 생성형 AI, 학생 인식, 프로젝트 기반 학습, 기여도

ABSTRACT

This study empirically analyzed learner perceptions and collaboration patterns in a university programming course where the use of Generative AI (GenAI) was formalized. The survey results indicated that while students highly valued the efficacy of GenAI during project execution, they simultaneously expressed concerns regarding dependency and fairness in traditional assessment contexts. Furthermore, an analysis of team project data over two years revealed that the variance in team contribution (Vg) decreased by 11.4% following the introduction of GenAI. Notably, a top-down normalization of collaboration was observed, as students previously in the free-rider range migrated into the moderate participation group. Based on these findings, this study discussed the necessity of foundational concept learning, strategic project design, multi-faceted PBL assessment frameworks, and the enhancement of learners' self-regulatory capabilities in the GenAI era. However, as a small-scale case study of a specific course, there are limitations to generalize the findings. Future research involving diverse learner groups and objective log data is required to further refine the analytical model.

Keywords Advanced Programming Education, Generative AI, Student Perception, Project-based Learning (PBL), Contribution Analysis

†정회원 경기대학교 소프트웨어경영대학
 AI컴퓨터공학부 교수(교신저자)

논문투고 2026년 02월 05일
심사완료 2026년 03월 17일
게재확정 2026년 04월 13일
발행일자 2026년 08월 04일

1. 서론

최근 대형 언어 모델의 급격한 발전은 대학 교육 전반에 대한 재검토를 요구하고 있다[1,2]. 이러한 변화는 컴퓨터 과학 전공 교육, 특히 프로그래밍 교육에서 더욱 두드러진다. 학생들은 학습과 문제 해결을 위해 생성형 AI(Generative AI) 도구를 적극적으로 활용하고 있다. 특히 2025년을 기점으로 생성형 AI의 능력이 급격하게 확장되고 초급 개발자에 근접한 수준의 지원이 가능해지면서, 대학의 프로그래밍 수업에서 언어 학습과 실습 중심의 틀은 근본적인 변화를 요구받고 있다.

한편 산업현장에서도 전공 졸업생에게 요구되는 역량은 변화하고 있다[3]. 단순한 코딩 능력에 대한 수요는 줄어들었고 알고리즘 설계, 아키텍처에 대한 이해, 최적화 및 문제 해결 능력까지 종합적인 능력이 더욱 중요해지고 있다. 이는 컴퓨터 과학 전공 교육이 보다 고차원적인 사고와 설계 능력을 기를 수 있도록 재구성되어야 함을 시사한다.

이러한 문제의식에 기반하여 본 연구진은 2학년 과정의 『객체지향 프로그래밍』 과목에 생성형 AI를 공식 도입한 수업을 설계하였다. 구체적인 AI 활용 지침 수립, 프롬프트 교육 시행, 팀 프로젝트를 통한 개방형 문제 해결 등 선행 연구에서 제안된 교수학습 전략들을 수업 설계에 적용하였다[4-6]. 본 연구는 해당 수업의 효과를 다면적으로 분석하여, 생성형 AI가 고급 프로그래밍 교육에서 갖는 가능성과 한계를 실증적으로 조명하고자 한다.

이러한 목적을 위해 연구 질문은 다음과 같다.

RQ1. 고급 프로그래밍 수업에서 생성형 AI 도입은 학생들의 학습 인식에 어떤 영향을 미치는가?

RQ2. 생성형 AI를 활용한 팀 프로젝트 과제는 학생들의 프로젝트 참여 양상과 협업 구조에 어떠한 변화를 보이는가?

RQ3. 생성형 AI를 활용한 프로그래밍 수업에서 기존 평가 방식의 한계는 무엇이며, 이에 대응하여 어떤 평가 방식의 변화가 필요한가?

위의 질문에 답하기 위해 본 연구에서는 AI 도입의 효과를 설문조사와 프로젝트 결과를 통해 분석하고자 한다. 먼저 AI를 도입한 수업 방식이 학생들의 학습 성취, 동기 부여, 자신감 등 학습 인식에 어떤 영향을 미치는지 설문 조사로 분석한다. 또한 팀프로젝트에서 AI 도입 효과를 구체적으로 파악하기 위해 해당 과목의 2024~2025년의 프로젝트 결과물과 학생 상호 평가 및 수업 후기를 비교 분석함으로써 협업 구조와 참여 양상의 변화를 정량적으로 살펴본다.

본 연구는 생성형 AI가 고급 프로그래밍 교육 현장에 미치는 파급효과를 학습자의 인식 변화와 참여 구조로 분석하고자 한다는 점에서 기존 연구와 차별성을 갖는다. 특히 프로젝트 중심의 통합적 문제 해결 과제에서, 생성형 AI 도입 전후의 변화를 실증적으로 비교함으로써 단순한 만족도 조사를 넘어선 구체적인 교육적 경험과 실행 사례를 제시하고자 한다. 이러한 분석은 AI가 일상화된 교육 환경에서 학습자의 실제적 성장을 어떻게 유도하고 평가할 것인가에 대한

경험적 근거를 제공할 것으로 기대된다.

2. 관련 연구

2.1 생성형 AI와 프로그래밍 수업의 변화

생성형 AI의 확산에 따라 학습자의 문제 해결 과정은 빠르게 변화하고 있으며, 프로그래밍 과목에서 그 영향이 가장 직접적으로 나타나고 있다. 이 분야의 선행 연구들은 크게 AI 활용 효과 분석, AI 활용 학습 시스템 개발, 그리고 교수학습 모델 제시라는 세 가지 축을 중심으로 전개되어 왔다. 본 연구에서 언급하는 AI는 학습자의 코딩 및 프로젝트 수행을 보조하는 생성형 AI를 의미하며, 문맥에 따라 혼용하여 기술하였다.

먼저, AI가 학습에 미치는 영향에 대한 다수의 연구는 AI가 학습자의 부담을 완화하고 문제 해결을 지원하는 유용한 도구임을 인정하면서도, 과도한 의존이 학습 과정을 약화시킬 수 있다는 우려를 동시에 보고해 왔다[6-9]. Rahe 등은 자체 개발 환경 내에서 AI 요청 유형을 분석한 결과, 별도의 교육적 장치가 없는 경우 AI 활용도가 높아질수록 의존도 역시 심화되는 현상을 확인하였다[10]. 이는 프로그래밍 학습에서 AI의 효과적인 도입을 위해 체계적인 가이드라인과 별도의 교수학습 체계가 필수적임을 시사한다.

이와 관련하여, 생성형 AI를 활용한 전용 학습 시스템 개발 연구도 활발하다. Li 등은 해당 제시가 아닌 학습 지원 역할을 수행하는 AI 활용 시스템을 개발하였으며[11], 박중오 등도 AI를 통합한 지능형 개발 환경을 통해 학생들의 학습을 지원하는 방법을 제안하였다[12]. 하지만 이러한 제한적 지원 기능을 탑재한 맞춤형 전용 시스템은 ChatGPT 등 범용 생성형 AI에 대한 접근이 일반화된 대학의 실제 학습 현장에 그대로 적용하기에는 한계가 있다.

교수학습 모델 측면에서도 도구 활용 방식에 따른 성과 차이가 확인되고 있다. Wang 등은 AI 도입이 체계적인 교수 설계가 필요하며 특히 정답 없는 열린 문제 해결 과제에 적합하다는 점을 지적하였다[5]. 한편 Ada 등은 AI 도입이 프로그래밍 진입 장벽을 낮추고 개념 이해도를 향상시키는 효과가 있음을 실증하였다[13]. 국내에서도 프로그래밍 수업에서 AI를 활용하는 교수 설계에 대한 연구가 활발하다[14, 15].

종합하면, 생성형 AI 도입의 효과와 한계를 고찰한 연구들이 많이 있었으나, 대부분 기술 도입 초기의 상황에 기반하고 있다. 따라서 고도화된 AI가 일상화된 2025년 이후의 학습 환경이 프로그래밍 교수학습 전반, 특히 팀 단위의 프로젝트 수행 방식에 어떠한 실질적인 변화를 가져왔는지에 대한 심층적인 연구가 요구된다.

2.2 프로젝트 기반 학습과 고차적 사고 역량

AI를 활용하는 수업에서 단순 지식 습득보다는 개방형·통합적 문제 해결을 강조하는 프로젝트 기반 학습(Project

Based Learning, PBL)의 효용성이 주목받고 있다[16-18]. PBL은 학습자가 문제 정의부터 설계, 구현, 검증의 전 과정을 주도하며 통합적 문제 해결 역량을 기르는 데 목적을 둔다. AI는 이러한 환경에서 기술적 장벽을 낮추고 학습자의 인지적 부하를 덜어주어, 복합적인 문제를 다룰 수 있게 할 것으로 기대된다. 특히 PBL에서 AI 활용을 통해 분석적 사고, 비판적 사고 및 협업 능력과 같은 고차적 사고 역량을 제고할 수 있을 것이라는 연구 결과가 있다[5].

한편 Guo 등은 고급 프로그래밍 과목을 재설계한 사례를 통해, AI를 지식 시스템으로 활용하며 코딩부터 최적화, 문서화까지 포함하는 개방형 다층적 프로젝트 모델을 제시하였다[16]. 특히 교수자가 단계별 과정을 모니터링하고 질문과 피드백을 통한 대면 평가를 병행함으로써, 통합적 다면 평가 체계의 사례를 보였다.

국내에서도 PBL과 AI의 결합에 관한 연구가 다수 발표되었으나 학습자 만족도나 인식 변화를 중심으로 논의하고 있다[17, 18]. 프로젝트 수행 과정에서 학생들의 AI 활용 형태나 역할 분담, 참여 구조의 변화에 대한 실증적 분석은 여전히 부족한 실정이다. 특히 대학의 고급 프로그래밍 수업에서 AI 도입이 프로젝트 수행에 미치는 영향에 대한 실증적 연구는 매우 제한적이다.

2.3 생성형 AI 활용 수업에서의 평가 방법의 변화

생성형 AI의 도입은 교수·학습 방식뿐만 아니라 평가 체계에도 새로운 과제를 던지고 있다. 기존의 프로그래밍 교육은 코드 실행 결과나 필기시험 중심의 평가가 위주였으나, AI가 코드 생성과 오류 수정까지 지원하게 되면서 제출된 결과물이 학습자의 실제 역량과 문제 해결 과정을 충분히 반영하기 어렵다는 연구 사례가 많이 발표되었다[7-9].

이러한 한계는 최근 연구에서도 확인된다. AI 활용이 문제 해결이나 과제 수행 속도는 높일 수 있으나, 필기시험이나 개별 코딩 시험 등 전통적 지표에서는 성취도 향상이 일관되게 관찰되지 않으며, 과도한 의존이 성적에 부정적 영향을 미친다는 보고도 있다[5, 9, 19]. 특히 AI 활용 경험이 학습자의 자신감을 과도하게 높여, AI 없이 문제를 해결해야 하는 실제 평가 상황과의 간극, 즉 인지부조화(Cognitive dissonance)를 초래할 위험이 있다[19]. 이는 학습자의 주관적 인식과 객관적 성취도 사이의 불일치를 유발하여 평가에 대한 혼란이나 불신으로 이어질 수 있다.

프로젝트 기반 수업과 평가가 이런 문제의 대안이 될 수 있으나 AI 활용을 전제할 때 프로젝트 평가도 학습 과정과 역할의 질적 변화를 포착하는 방향으로 재편되어야 한다. 이를 위해서 교수자의 현실적인 교육 여건을 고려한 구체적인 평가 사례가 계속 축적되어야 할 것이다.

2.4. 본 연구의 차별성

이상의 선행연구 검토 결과, 생성형 AI는 프로그래밍 교육에서 무시할 수 없는 핵심 도구로 부상하고 있음을 알 수

있다. 그러나 기존 논의는 주로 학습 만족도와 같은 단편적 지표에 치중되어 있어, 실제 교육 현장의 복합적인 변화를 포착하는 데 한계가 있다.

본 연구는 크게 세 가지 측면에서 기존 연구와의 차별성을 확보하고자 한다. 첫째, AI 도입 효과의 측면에서 학습, 개별 과제, 팀 프로젝트 등 과업의 성격과 맥락에 따른 학습자 인식의 분화를 심층 분석한다. 둘째, AI 활용이 팀 프로젝트 내의 실질적인 참여 구조와 협업 방식에 미치는 영향을 실증적으로 규명한다. 셋째, 실제 사례를 통해 기존 평가 방식의 한계와 변화 방향을 살펴본다.

결론적으로 본 연구는 AI 활용이 공식화된 고급 프로그래밍 수업을 대상으로 학습자 인식의 맥락적 차이와 협업 구조의 변화, 그리고 다면적 평가 결과를 입체적으로 분석함으로써, AI 활용 프로그래밍 교육의 실효성 있는 교수 설계 및 평가 체제 구축을 위한 경험적 근거를 제공하고자 한다.

3. 수업 설계와 연구 방법

이 장에서는 앞서 제시한 연구 질문을 검토하기 위해 본 연구에서 사용한 수업 설계와 자료 수집 및 분석 방법을 설명한다. 본 연구는 단일 대학의 고급 프로그래밍 과목을 대상으로 한 사례 연구로서, 생성형 AI 활용을 공식화한 수업 환경에서 학생들의 학습 인식과 프로젝트 참여 양상을 다각도로 살펴보고자 하였다. 이를 위해 수업 운영 방식과 AI 활용에 대한 교수·학습 설계를 먼저 제시한 뒤, 학기 중 실시된 설문조사 자료와 팀 프로젝트 수행 결과, 그리고 개별보고서에 포함된 동료 평가 데이터를 분석 대상으로 활용하였다. 이러한 복합 자료 분석을 통해 생성형 AI 활용 수업이 학생들의 인식과 프로젝트 수행 과정에 어떠한 특징적인 변화를 보였는지를 탐색한다.

3.1 연구 맥락 및 대상 과목 개요

본 연구는 연구진이 근무하는 대학의 전공 교과목인 『객체지향 프로그래밍』 수강생 150명(총 3개 분반)을 대상으로 수행되었다. 해당 과목은 자바(Java) 언어를 기반으로 인터페이스, 제너릭, 컬렉션, Object 클래스, 박싱(Boxing) 등 객체지향 핵심 설계 원리를 이해하고 실제 소프트웨어개발 능력을 함양하는 것을 목표로 한다.

본 교과목의 구체적인 수업 운영 및 평가 체계는 Table 1과 같다. 수업은 지식 전달의 효율성을 높이기 위해 플립드러닝(Flipped Learning) 방식을 채택하였으며, 학생들은 온라인 동영상 강의로 이론을 선행 학습한 후 대면 수업에서 교수자의 가이드에 따라 예제 심화 실습을 수행한다.

특히 중간고사 이후 진행되는 6주간의 팀 프로젝트는 생성형 AI 도입에 따른 협업 방식 변화를 분석하기 위한 핵심적인 데이터를 제공한다.

Table 1. Characteristics and Components of the Course

Category	Key Details
Course Description	Object-Oriented Programming (Java-based)
Target & Enrollment	2nd-year CS majors (150 students, 3 classes)
Instructional Method	Flipped Learning (Online: Theory / In-class: Practice)
Assignments (10%)	Application development tasks (5 times total)
Examinations (55%)	Midterm/Final written (50%), Practical exam (5%)
Team Project (10%)	6-week team project (Full development cycle)

Table 2. Generative AI Utilization Guidelines (2025)

Category	AI Integration Strategies
Learning	Prompt engineering training & AI as a learning scaffold
Assignments	Phase 1: Structure generation using AI / Phase 2: Extension & Code review
Project	Full-cycle AI use, utilization reports
Assessment	Written/Practical exams: No AI allowed (Independent verification)

3.2 생성형 AI를 활용한 수업 설계

본 연구의 수업 설계 핵심은 학습 과정 전반에 생성형 AI 활용을 명시적으로 허용하고 장려하였다는 점이다. 2024년과 2025년 수업은 교과 내용, 평가 비중, 프로젝트 기술적 난이도 등 대부분의 조건을 동일하게 유지하였고, 2025년 수업에서는 생성형 AI 활용을 위해 Table 2와 같이 안내하였다.

먼저, AI 활용 가이드라인 수립 및 사전 교육이다. 학생들에게 AI가 생성한 코드를 무분별하게 복제하는 대신, 코드의 구조를 이해하고 스스로 확장하는 2단계 과제 설계와 활용 지침을 제공하였다. 학기 초에는 효율적인 프롬프트 작성 방법과 결과물 검증에 대한 기초적인 안내를 실시하여 AI 활용의 심리적 장벽을 낮추고 오용을 방지하고자 하였다.

팀 프로젝트는 자바 GUI(Swing) 기반 애플리케이션 개발을 주제로 진행하였다. 2024년(게임 콘텐츠)과 2025년(데이터 기반 CRUD 시스템)의 프로젝트 주제는 다르지만, 객체지향 설계 원칙 준수와 데이터 분리라는 기술적 요구 사항은 동일하게 유지하였다.

팀 프로젝트의 전 과정에서 생성형 AI의 활용을 장려하였다. 학생들은 전반기 수업에서 AI에 대한 활용법을 습득한 상태로 프로젝트에 임하게 되며, 이러한 수업 설계는 생성형 AI 활용을 전제로 학생들의 참여와 협업 양상을 관찰하기 위한 목적에 부합할 것으로 기대된다. Table 3은 프로젝트 수행 방법과 평가 개요를 보여준다.

Table 3. Team Project Structure and Design

Category	Requirements & Evaluation
Technical Requirements	Adherence to OOP principles, File-based I/O implementation, Java Swing GUI
Execution & Review	Mid-term progress feedback, Final oral presentation & demonstration
Evaluation Criteria	Team: Completeness, Collaboration Individual: Peer review, Reflection paper

3.3 설문조사 연구 자료 및 수집 절차

본 연구의 설문조사는 생성형 AI 활용 환경에서 학생들의 프로그래밍 효능감, AI 활용 인식, 그리고 과제 및 프로젝트 수행 시 AI의 도움 정도가 수업 진행에 따라 어떻게 변화하는지 탐색하기 위해 설계되었다.

설문은 학기초(T1), 중반(T2), 학기말(T3) 총 3차에 걸쳐 전체 수강생 대상으로 실시되었으며 프로그래밍 자신감과 학습, 과제, 프로젝트 수행 과정에서의 AI 도움에 대한 경험 및 인식을 공통 문항으로 포함하여 인식 변화의 경향을 살펴보고자 하였다. 이와 함께 생성형 AI 활용에 따른 의존성 및 평가 공정성에 대한 우려 문항을 중간(T2) 및 학기말(T3) 설문문에 포함하였다.

본 연구에서 학생들의 인식 변화를 측정하기 위해 사용된 설문 도구는 기존 연구의 설문 도구를 본 연구 맥락에 맞게 수정 보완하였다[7, 9]. 본 연구에서는 차수별로 동일집단을 대상으로 하였으며 T3는 T2 설문 응답자를 대상으로 하였다. 그러나 응답자 수가 상이한 점을 고려하여 개인별 변화 추적보다는 차수별 응답 분포와 전체적인 경향 분석에 초점을 두었다.

Table 4는 설문 차수별로 조사한 주요 측정 영역과 문항 구성을 요약한 것으로, 동일 개념을 반복 측정하여 인식 변화의 흐름을 파악할 수 있도록 구성하였다. 전체 설문 문항 및 문항별 응답 결과는 부록의 표 10 ~ 15에 첨부하였다.

Table 4. Survey Composition and Measurement Timeline

Category	Key Metrics (Sample Items)	T1	T2	T3
Programming Efficacy	Confidence in Java and OOP concepts	●	●	
AI Competency	Proficiency and confidence in using GenAI	●	●	●
Perceived AI Support	AI contribution to learning, coding, and debugging	●	●	●
Learning Background	Prior coding experience and AI usage frequency	●		
Learning Attitude	Programming preference and problem-solving style	●		
AI Risks/Ethics	Concerns about AI dependency and fairness		●	●
Team Project Experience	AI contribution to collaboration and role changes			●
Number of Respondents		142	67	33

3.4 팀프로젝트 연구 자료 및 수집 절차

이 수업에서는 매년 중간고사 이후 팀프로젝트를 운영하였으며, 연구진은 다년간의 팀프로젝트 운영 데이터와 경험을 축적하고 있다. 본 연구에서는 이를 이용하여 2025년 팀프로젝트의 학습자 참여 양상과 협업 구조의 변화를 탐색하고자 한다. 2025년을 기점으로 학생들의 생성형 AI 활용 정도나 수준이 많이 변화하였으며, AI 활용을 바라보는 사회적 산업적 인식도 크게 변화되었다. 이렇게 기술적 사회적 혁신의 폭이 컸던 2025년의 상황을 가늠하기 위하여 전년도 동일과목 운영과 비교하여 이 수업에서 변화한 상황과 의미를 확인하고자 하였다.

이를 위하여 해당 수업의 2024학년도와 2025학년도 팀프로젝트 자료를 비교 분석하였다. 두 연도의 과목 구성과 프로젝트 주제, 평가 기준은 대체로 유사하였으며, 주요 차이는 생성형 AI 활용이 2025학년도에는 공식적으로 허용·장려되었다는 점이다.

팀프로젝트는 매년 반 별로 7-8개 팀으로 구성되며 각 팀의 인원은 6~8명 정도이다. 분석 대상 보고서는 팀의 상대평가 결과 상중하 각 4/8/3개씩 총 15개 팀의 팀 및 개별보고서를 이용하였다. 이는 평가결과의 분포에 맞춰 균형 있게 분석하기 위한 것이다. Table 5에서 보듯이 본 연구에서 검토한 개별보고서는 년도별로 2024년 84건, 2025년 107건이며, 2025년에는 수강인원과 팀규모가 전년 대비 증가하였다.

팀 프로젝트 결과로 제출된 자료는 팀별 프로젝트 결과물, 개별 학생이 제출한 개별보고서(소감 및 팀원 간 상호 기여도 평가) 자료이다. 특히 개별보고서의 기여도 평가는 팀원들이 프로젝트 수행 과정에서 각 구성원의 기여도를 본인 제외하고 백분율(%)로 배분하는 상대평가 방식으로 이루어졌다. 기여도 점수에 대해 사전에 개인별 성적 평가에 반영됨을 안내하여, 신중한 평가를 유도하였고 각 팀원에 대해 간단한 역할을 같이 기술하게 하여 점수의 신뢰도를 높였다.

Table 5. Data Collection and Analysis Overview

Category	2024	2025	Remarks
Total Students	128	152	3 classes
Total Teams	24	23	Team evaluation criteria: Completeness, Collaboration
Analyzed Teams	15	15	Selected by performance distribution
Analyzed Reports	84	107	Individual project reports
Avg. Team Size	5.6	7.1	All members in analyzed teams

본 연구에서는 팀 프로젝트의 참여 구조를 분석하기 위해 팀원 수에 따른 차이를 보정하는 방법으로 개인 기여도 평가 점수를 정규화하였다. 팀원수가 N일 때 각 팀원에 대해 N - 1명이 상호 평가한 기여도 점수의 개인별 평균값(C_i)을 산출하였다. 그리고 각 팀에서 모든 구성원이 동일하게 기여한 경우의 $100 / (N - 1)$ 을 기준값으로 설정하고, 이를 기준으

로 개인 i의 정규화된 기여도를 다음과 같이 정의한다.

$$\text{Relative Contribution } RC_i = \frac{C_i - 100/(N-1)}{100/(N-1)}$$

where C_i is an average personal score and N is a team size.

RC_i 는 팀 내 상대적 기여 위치를 나타내며, 0에 수렴할수록 균등 기여를 의미한다. 본 수식은 팀 규모(N)에 따른 기여 밀도를 반영하여 정규화되며, 이론적으로 모든 팀원으로 부터 독보적인 평가를 받을 경우 양(+)의 최댓값을, 기여도가 전무할 경우 -1에 근접하게 된다. 이는 팀 규모와 관계없이 기여도의 편차를 객관적으로 비교 가능하게 한다.

또한, 팀 내 기여도가 특정 인원에게 편중되거나 균등한 지를 측정하기 위한 척도로 팀별 기여도 분산을 산출하였다. 이 지표는 개별 팀원들의 정규화된 기여도 지수 RC_i 가 산술적 평균인 0으로부터 이탈한 정도를 나타낸다. 팀별 기여도 분산(V_g)의 계산식은 다음과 같다.

$$\text{Team Contribution Variance } V_g = \frac{\sum_{i=1}^N (RC_i)^2}{N}$$

Table 6은 프로젝트 기여도 점수의 취합과 처리 과정을 예시와 함께 보여준다. (부록 표 16 ~ 17 참조)

팀별 기여도 분산값 V_g 는 팀 내의 협업 불균형 정도를 나타내는 핵심 척도로 활용된다. V_g 값이 0에 가까울수록 팀원들이 역할에 따라 균등하게 참여한 수평적 협업 구조임을 의미하며, 반대로 이 값이 클수록 특정 인원의 '독주 현상'이나 '무임승차 현상'이 강하게 나타나는 불균형한 참여 구조임을 시사한다. 팀 내 기여도의 편중 정도를 측정하기 위해 분산(V_g)을 지표로 활용하는 방식은 조직 및 교육학 연구에서 자원이나 영향력의 불균형을 측정하는 방법론으로 제안되었다[20].

Table 6. Procedure for Calculating Team Contribution Metrics

No	Task Process	Output Example																																				
1	Raw Score Collection	Initial contribution scores (Peer)																																				
2	Team Contribution Matrix N x (N-1) matrix	<table><tr><td>0</td><td>30</td><td>25</td><td>15</td><td>17</td><td>13</td></tr><tr><td>20</td><td>0</td><td>35</td><td>15</td><td>15</td><td>15</td></tr><tr><td>20</td><td>35</td><td>0</td><td>15</td><td>16</td><td>14</td></tr><tr><td>18</td><td>30</td><td>25</td><td>0</td><td>15</td><td>12</td></tr><tr><td>17</td><td>32</td><td>23</td><td>15</td><td>0</td><td>13</td></tr><tr><td>15</td><td>35</td><td>25</td><td>14</td><td>11</td><td>0</td></tr></table>	0	30	25	15	17	13	20	0	35	15	15	15	20	35	0	15	16	14	18	30	25	0	15	12	17	32	23	15	0	13	15	35	25	14	11	0
0	30	25	15	17	13																																	
20	0	35	15	15	15																																	
20	35	0	15	16	14																																	
18	30	25	0	15	12																																	
17	32	23	15	0	13																																	
15	35	25	14	11	0																																	
3	Individual Contribution Average C_i	<table><tr><td>18.0</td><td>32.4</td><td>26.6</td><td>14.8</td><td>14.8</td><td>13.4</td></tr></table>	18.0	32.4	26.6	14.8	14.8	13.4																														
18.0	32.4	26.6	14.8	14.8	13.4																																	
4	Relative Contribution RC_i	<table><tr><td>-0.10</td><td>0.62</td><td>0.33</td><td>-0.26</td><td>-0.26</td><td>-0.33</td></tr></table>	-0.10	0.62	0.33	-0.26	-0.26	-0.33																														
-0.10	0.62	0.33	-0.26	-0.26	-0.33																																	
5	Contribution Variance V_g	$V_g = 0.1246$																																				

이러한 정규화 지수(RC_i)와 팀별 분산 지표(V_g)는 팀원 수라는 변수를 통제된 상태에서 협업의 질적 구조를 정량화할 수 있다. 본 연구에서는 이 지표를 활용하여 2025년의 프로젝트에서 전년도와 비교하여 기여도의 양극화 현상이 완화된 것인지 여부를 규명하고자 한다.

3.5 연구의 범위와 한계

본 연구는 단일 전공 교과목을 대상으로 수행된 사례 연구로서, 모든 변인이 엄격히 통제된 실험 환경이 아닌 실제 교육 현장의 복합적인 역동성을 내포하고 있다. 따라서 연구 결과 해석에 있어 다음과 같은 한계를 가진다.

먼저, 협업 도구 및 기술 생태계의 비약적인 변화이다. 2024년과 2025년 사이 발생한 생성형 AI 모델의 성능 고도화와 답변 품질의 개선, 그리고 GitHub 등 협업 도구에 대한 학습자의 숙련도 변화는 팀 프로젝트의 효율성에 복합적인 영향을 미쳤을 가능성이 크다. 또한, 기술 발전에 따른 사회적 관심도의 증가가 학습자의 동기 부여와 자기효능감에 미친 간접적 영향 역시 배제하기 어렵다. 그러나 이러한 종합적인 교육 환경의 변화를 포함하여 이를 객관적 지표의 추세를 통해 확인하는 것은 현장 중심 연구로서 중요한 학술적 의미를 지닌다.

아울러 학기 중 수행된 설문 조사는 차수별 응답자 수의 편차로 인해 결과를 일반화하는 데 주의가 필요하다. 본 연구에서 설문 결과는 정량적 분석을 보완하는 보조적 도구로 활용되었으며, 과제·프로젝트·평가 등 다면적인 인식을 교차 분석함으로써 프로그래밍 교육 내 AI 도입의 가능성과 잠재적 우려를 입증하는 데 주된 목적을 두었다.

4. 결과 분석

본 장에서는 생성형 AI 활용이 공식화된 고급 프로그래밍 수업에서 나타난 학생들의 인식 변화와 프로젝트 수행 양상을 분석한다. 이를 위해 설문조사 결과와 팀 프로젝트 자료를 바탕으로, AI 활용 수업 환경에서 학생들이 어떻게 학습을 인식하고 프로젝트에 참여하였는지를 다각도로 살펴본다.

구체적으로는 먼저 설문조사 결과를 통해 AI 활용에 대한 학생들의 인식과 자가진단 경향을 정리하고, 이어서 2024년과 2025년 팀 프로젝트 자료를 비교하여 프로젝트 참여 구조와 기여도 분포의 변화를 분석한다. 마지막으로 다면적 평가 사례와 이에 대한 학생 인식을 보완적으로 고찰한다.

4.1. 설문조사 결과: AI 활용 인식과 자가진단 경향

본 절에서는 생성형 AI 활용 수업에 참여한 학생들의 인식 변화와 자가진단 경향을 분석한다. 본 연구의 설문은 새로운 수업 환경 속에서 학습자가 체감하는 심리적·태도적

변화를 파악하는 데 중점을 두었다. (설문 결과는 부록 표 10~15 참조)

4.1.1. 초기 학습자 특성 및 AI 활용 경험

학기 초 실시된 사전 설문 결과, 수강생들은 프로그래밍 경험과 자신감 측면에서 큰 편차를 보였다. 자바 언어를 처음 접하는 학생(19%)부터 웹 개발 및 AI 활용 경험을 보유한 학생까지 분포가 다양하였으며, 프로그래밍 과제 수행 및 디버깅에 대한 자신감은 긍정 응답과 부정 응답이 각각 30%로 비슷하게 나타났다.

다만, 중간 설문에서 나타난 학습 성향 조사에서는 과제 수행의 몰입도(95%), 테스트 과정의 즐거움(76%), 코드 품질 향상 의지(74%) 등에서 대다수가 긍정적으로 응답하였다. 이는 수강생들이 기본적인 프로그래밍 학습 동기와 역량을 갖추고 있음을 시사한다.

반면, 학기 초 생성형 AI 활용 경험은 비교적 제한적이었다. AI를 자주 활용한다는 응답은 35% 수준이었으며, 전혀 경험이 없는 학생도 11%에 달했다. 프롬프트 작성 역량(28%)이나 AI 생성 결과의 수정·통합 능력(42%)에 대한 자가진단 역시 과반에 미치지 못하여, 학기 초에는 약 3분의 1 정도의 학생만이 실제 프로그래밍에 AI를 활용한 경험이 있었음을 알 수 있다.

4.1.2. 개별 과제와 팀 프로젝트에서 인식의 분화

조사 결과, 생성형 AI에 대한 학생들의 인식은 학습의 맥락에 따라 뚜렷한 차이를 보였다. 우선 학습 보조 도구로서의 AI는 긍정적인 평가를 받았으나, 이를 개별 과제나 시험에 활용하는 것에 대해서는 유보적이고 신중한 입장이 관찰되었다.

과제 수행 시 AI 활용을 선호한다는 응답은 71%였으나, AI가 생성한 결과 코드 사용에 대해서는 58%만이 긍정적으로 답했다(부록 <표 13>). 특히 기초 프로그래밍 수업에서의 AI 사용이나 평가 상황(시험)에서의 활용은 부정적인 의견이 65.7%로 높게 나타났다(95% CI [54.3%, 77.1%])(부록 <표 15>). 이는 학생들이 AI 활용으로 인한 의존성 심화와 논리적 사고력 저하를 경계하고 있음을 보여준다. 실제로 응답자의 상당수가 AI 의존성(47%) 및 알고리즘 구상 노력의 감소(37%)를 주요 우려 사항으로 꼽았다(부록 <표 13>).

그러나 팀 프로젝트 수행과 관련해서는 AI 활용에 대해 긍정적인 인식이 더 높았다. 프로젝트 수행에 AI가 도움이 되었다는 응답과 AI 없이는 수행이 어려웠을 것이라는 응답은 모두 89%로 매우 높게 나타났다(95% CI [82.0%, 96.8%]). 특히 AI 활용을 통해 개발 자신감을 얻었다는 가라는 질문에 부정 응답이 전무했다는 점은 주목할 만하다. 도움을 받은 구체적 영역으로는 문제 해결(30%), GUI 코드 생성(18%) 외에도 모든 면에서 도움이 되었다(37.5%)는 응답이 높게 나타나(부록 <표 15>), 복합적인 개발 과정 전반에서 AI의 효용성을 체감했음을 알 수 있다. Table 7은 개별

과제와 팀프로젝트에 대한 학생 인식의 차이를 요약한다.

Table 7. Comparison of Perceptions on GenAI Use by Task Type

Category	Individual Tasks & Exams	Team Projects
Context	Knowledge acquisition & Evaluation (Closed-ended)	Complex problem solving (Open-ended)
Primary Attitude	Conservative & Cautious	Proactive & Receptive
Positive Response	Approval for use: 71.6%	Perceived helpfulness: 89.4%
Negative / Concern	Opposition to use in basic courses: 66.7%	Difficulty performing without AI: 63.6%
Primary Support Area	Error debugging (51%), Concept explanation (15%)	Problem solving (30%), All stages (37.5%)
Key Concerns	Increased dependency, Reduced thinking skills	

4.2 팀 프로젝트 기여도 및 협업 구조 분석

본 절에서는 생성형 AI 활용이 공식화되기 전(2024년)과 이후(2025년)의 팀 프로젝트 자료를 비교하여, 팀 내 기여 구조와 참여 양상의 변화를 분석한다. 본 절의 분석은 실제 프로젝트 수행 과정에서 나타난 참여 구조를 실증적으로 살펴보는 데 목적이 있다.

4.2.1 정량적 분석: 개인 기여도 분포

학습자의 프로젝트 참여도가 2025년에 어떤 변화가 있었는지 확인하기 위해 개별 기여도 값을 이용하였다. 개별 기여도를 나타내는 RC_i 지표는 팀 평균 기여도를 0으로 기준화한 값으로, $RC_i \geq 0.5$ 는 해당 학습자가 팀 평균의 1.2배 이상의 기여를 한 것을 의미하며, $0 \sim 0.5$ 구간은 1명 이상의 기여를 한 구간이다. 한편 $-0.5 \sim 0$ 은 부분 기여자로 해석할 수 있는데, 여기서 -0.5 는 평균의 팀규모에 따라 다르지만 85% 수준을 수행한 참여기준으로, 저기여자 구간을 나누는 임계값은 Bobko 등의 4/5 규칙을 참고하였다[21].

구간에 따른 연차별 기여도 분포를 비교 분석한 결과는 Table 8과 같다. 2025년 생성형 AI 도입 이후 팀 프로젝트 내 참여 양상에 유의미한 구조적 변화가 포착되었다. 팀 내 주도적 역할을 수행하는 집중 기여자 구간($RC_i \geq 0.5$)의 비율은 2024년 7.1%에서 2025년 6.5%로 소폭 감소하였다. 그런가 하면 최하위 구간의 비중은 2024년 9.5%에서 2025년 4.7%로 절반 이하로 급감하였다. 최하위의 저기여자 구간은 프로젝트에 거의 참여하지 않았거나 실질적인 기여가 확인되지 않은 무임승차로 볼 수 있다.

한편 가장 두드러진 변화는 부분기여 구간($-0.5 < RC_i < 0$)의 학생 비율이 40.4%에서 52.3%로 약 12%p 증가한 점이다. 이는 과거 프로젝트 참여에 소극적이었던 하위권 학생들이 중간 참여층으로 대거 진입했음을 보여준다고 할 수

있다.

요약하면 2024년 대비 2025년 프로젝트에서는 저기여 구간의 인원이 감소하고 부분 기여 구간의 비중이 확대되는 양상이 확인되었다. 이는 팀 내 기여가 특정 소수에게 편중되는 구조적 특징은 여전하지만, 과거 비참여 상태에 머물렀던 하위 기여자들이 일정 수준 이상의 과업을 수행하는 방향으로 참여 분포가 재편되었을 가능성을 보여준다.

Table 8. Distribution of Individual Contribution (RC_i) by Performance Range

Contribution Range	Interpretation	2024 (%)	2025 (%)	Change(%p)
$0.5 \leq RC_i$	Leaders	7.1	6.5	-0.6 ↓
$0 \leq RC_i < 0.5$	Moderate Contributors	42.9	36.4	-6.5 ↓
$-0.5 < RC_i < 0$	Partial Contributors	40.4	52.3	11.9 ↑
$RC_i \leq -0.5$	Low Contributors (Free-riders)	9.5	4.7	-4.8 ↓

4.2.2 정량적 분석: 팀별 기여도 분산

이러한 분포 변화는 팀별 기여도 분산(V_g) 지표를 통해서도 드러난다. 3.4.1절에서 정의한 바와 같이, 팀내 기여도 분산 값이 크다는 것은 특정 인원에게 과업이 집중되거나 일부 인원의 참여가 거의 없는 불균형한 협업 구조를 의미한다. 팀별 분산값의 분석 결과는 Table 9와 같다. (부록 <표 18> 참조)

Table 9. Intra-team Variance Summary

Contribution Range	2024 (w/out AI)	2025 (with AI)	Change
Average Intra-team Variance (V_g)	0.1217	0.1078	11.4% ↓
Maximum Variance Team	0.353	0.292	17.2% ↓
Standard Deviation (SD) of V_g	0.1169	0.0844	27.8% ↓
Ratio of Low-Contribution Segment ($-0.2 \leq RC_i < 0$)	21.4%	30.8%	11.9%p ↑

N = 15

p = 0.713, CI = 95%, Cohen's d = 0.136

분석 결과, 팀별 기여도 분산(V_g)의 평균값은 2024년 0.1217에서 2025년 0.1078로 전년 대비 약 11.4% 감소하였다. 소규모 표본의 특성상 통계적 유의성은 낮으나 ($p=0.713$), 분산의 표준편차가 0.1169에서 0.0844로 크게 낮아진 점은 주목할 만하다. 이는 2024년에 비해 2025년의 팀 프로젝트 협업 구조가 안정화되고 상향 평준화되었음을 의미한다.

팀별 분산이 줄어든 효과는 Fig. 1의 분포 곡선에서 분명하게 드러난다. Fig. 1은 2024년과 2025년 팀들의 분산 값을 내림차순으로 정렬하여 비교한 결과이다. 기여도 평준화

효과는 Fig. 1의 2025년 분포 곡선이 전년도에 비해 완만해지고 전체적인 수치가 하향 안정화된 것을 보여준다. 구체적으로는 2024년에는 분산이 0.3을 초과하는 독주형 팀이 2개 존재했으나, 2025년에는 최상위 불균형 팀들의 분산이 크게 감소하며 전체적으로 평준화되는 양상을 보였다.

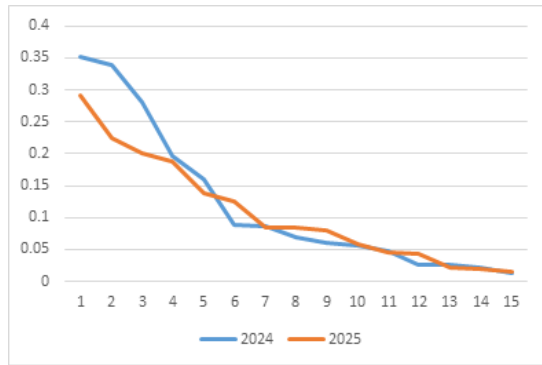


Figure 1. Comparison of Team Contribution Distributions (Decreasing Order Each Year) 부록 <표 18> 참조

종합하면, 2025년에는 전년도에 비해 팀 프로젝트 내의 고질적인 기여도 불균형이 완화되고 협업 구조가 평준화된 것으로 보인다. 팀별 기여도 분산(Vg)의 평균이 11.4% 감소하고, 분산의 표준편차(SD) 또한 27.8% 줄어든 결과는 2025년에 팀 간 편차가 줄어들어 협업의 양상이 전반적으로 안정화되었음을 방증한다. 더불어 극단적 저기여자의 비중이 줄어들고 상대적 저기여 구간의 비율이 약 12%p 증가한 점은, AI가 기술적 진입 장벽을 낮추어 기존의 무임승차 인원을 실질적인 참여층으로 전환시켰을 가능성을 보여준다.

4.2.3 협업 양상에 대한 질적 분석

정량적 기여도 분석 결과가 실제 현장에서 어떻게 나타났는지 확인하기 위해 핵심 기여자의 개별 보고서와 교수자의 관찰 기록을 분석하였다.

2024년 팀별 고기여 학생들의 보고서에서는 팀원 간 역량 차이로 인한 리더의 물리적 과부하가 두드러졌다. 팀원이 “정리되지 않은 코드를 제출해 다시 짜야”했다거나, “통합하는 데 너무 많은 시간을 뺏”겼다는 등의 기술은 리더에게 가중된 통합 부담을 단적으로 보여준다. 반면, 2025년에는 상대적으로 이러한 불만이 적었고 특히 GitHub의 활용도가 높아져서 다수가 직접 참여하는 구조로 바뀐 것을 확인할 수 있었다¹⁾. 거듭되는 저장소 재구축 등의 시행착오가 관찰되었으나, 전년도에 비해 참여가 확대되는 방향으로 변화하는 현상이 뚜렷했다.

이러한 변화는 교수자의 직접적인 관찰 결과와도 일치한다. 동일 과목을 연속 담당한 교수자의 관찰에 따르면, 2025년 프로젝트 결과물은 이전 연도와 비교하여 표현의 다양성과 구현의 안정성 측면에서 상당히 개선되었다. 기술적 환경(Swing GUI 기반)이 동일했음에도 불구하고 시각적·기능적 완성도가 향상되었고, 시연의 안정성과 완성도도 높아

졌다. 또한, 교수자의 중간 피드백의 수용도와 반영률이 높아졌다. 이것은 이전 년도에 비해 AI의 도움으로 결과물의 구축 속도가 빨라지면서 내용과 완성도가 풍부해진 것으로 보인다.

결과적으로 이전 년도에 비해 2025년에는 팀원들이 프로젝트의 전체적인 완성도와 설계의 본질에 집중할 수 있었는데, 이는 팀원 간의 마찰이 완화되고, 보다 생산적이고 창의적인 협업 환경이 조성된 것으로 평가할 수 있을 것이다.

4.3. 생성형 AI 활용에 따른 교수학습 설계 및 평가

앞선 설문조사와 프로젝트 분석을 통해 생성형 AI 활용에 대한 전반적인 긍정적 인식을 확인하였다. 그러나 자유 응답과 강의평가 의견을 종합해 볼 때, 학습자의 반응은 단일하지 않고 학습 맥락과 과업의 성격에 따라 뚜렷하게 분화되어 나타났다. 본 절에서는 평가 방식에 대한 인식 차이, 그리고 강의평가에 나타난 학생들의 의견을 고찰한다.

과제 수행과 평가에 대한 인식의 분화를 살펴보겠다. 과제 수행 시 AI의 도움에 대해서는 긍정적이었으나(68.7%, CI 95% [60.8%, 76.5%]), 일부 유보적인 입장(9%)도 존재했다. 특히 단계별 과제 설계와 설명을 포함한 보고서 제출을 의무화했음에도 불구하고, 상당수 학생이 보고서 작성 단계까지 AI를 활용한 것으로 파악되었다. 이는 정답 도출이나 지식 습득 여부만을 평가하는 기존 방식이 AI 활용 환경에서는 제약을 가질 수밖에 없음을 보여준다.

한편, 기초 과목 도입이나 시험에서의 AI 활용에 대해서는 부정적 혹은 유보적 태도가 주를 이루었다. 이는 프로그래밍의 기초 개념 형성과 개인 역량 검증 단계에서는 AI의 개입이 학습 효과 저하 및 평가 공정성 훼손으로 이어질 수 있다는 학생들의 우려를 반영한다. 결과적으로 학습자들은 학습 단계와 평가 목적에 따라 AI 활용의 경계를 구분하여 인식하고 있었다.

한편 전통적 평가 체제와의 간극이 뚜렷이 나타났는데, 평가 방식에 대한 학생 인식은 강의평가에서 두드러졌다. AI를 활용한 학습 경험으로 인해, 학생들은 지필시험(암기 및 손코딩 중심)에 대해 혼란과 부담을 토로했다. 이는 AI를 활용한 역동적인 개발 경험과 정적인 평가 방식 사이의 인식적 괴리를 보여준다.

종합하면, 학생들은 AI 활용에서의 평가에 대해 과제 유형과 목적에 따라 수용도나 공정성에 대한 견해의 편차가 존재했다. AI 도입은 프로그래밍 교육 현장에서 학습자의 인식 변화와 팀 프로젝트 내 참여 구조의 재편을 가져오는 것으로 보인다. 그런가 하면 전통적인 평가 방식과 AI 기반의 학습 경험 사이에는 여전히 상당한 간극이 존재한다. 이에 다음 장에서는 본 장에서 도출된 실증적 근거들을 바탕으로, AI 시대에 부합하는 프로그래밍 교육의 방향성과 지속 가능

1) GitHub의 활용은 2024년에 개별적인 일부 사용에 그쳤던 것에 비해 2025년에 사용이 전체 팀에서 일반화되었다. 이것은 기술의 확산과 함께 AI의 활용에 의해 기술 적용 허들이 낮아진 것도 한 원인으로 보인다.

한 평가 체제 구축을 위한 함의를 논의하고자 한다.

5. 연구 결과 고찰

5.1 프로그래밍 수업에서 AI 활용의 양면성과 교수학습의 과제

2025년은 생성형 AI가 교육 현장에 본격적으로 도입되며 교수학습 패러다임의 근본적인 변화를 몰고 온 한해였다. Becker 등은 AI 시대의 프로그래밍 교육은 AI가 내놓은 결과를 비판적으로 검토하고, 복합적인 문제를 AI가 이해할 수 있는 단위로 분해하며, 이를 다시 전체 시스템으로 통합하는 조율자(Orchestrator)를 양성해야 한다고 표현했다[22].

본 연구의 설문 결과는 AI가 이미 학생들에게 일상의 학습 도구로 정착했음을 보여준다. 학기 초 AI 활용 경험이 전무했던 학생들조차 단기간에 높은 효능감과 자신감을 보인 점은, 프로그래밍 교육에서 AI 활용의 필요성이 높음을 시사한다.

그러나 학습자의 높은 수용성 이면에는 해결해야 할 난제들이 많이 있다. 과제에서 상당수 학생이 코드뿐만 아니라 보고서까지 AI에 의존하여 제출하였다. 또한 AI 도움 없는 지필평가와 같은 전통적 체제에서 학생들이 느끼는 무력감은, 변화된 학습 환경과 정적인 평가 방식 사이의 깊은 인식적 괴리를 드러낸다. 학생들도 AI 의존 심화로 인한 문제 해결의 동기과 끈기 저하를 우려하고 있다는 점은 주목할 만하다.

최근 교육용 AI를 적용하여 정답 대신 가이드를 제공하는 방식의 시스템 개발이 활발히 논의되고 있으나[10-12], 범용 AI 접근이 자유로운 대학 환경에서 학습자의 도구 활용을 강제로 제약하는 방식은 실효성에 한계가 있다.

다만 본 연구에서 확인했듯, 단순 과제보다 프로젝트 수행 단계에서 학생들이 보여준 높은 AI 효용감과 긍정적 태도는 고무적이다. 이는 AI가 단순 지식 습득보다 고차원적 사고와 복합적 문제 해결이 요구되는 과업에서 더욱 효과적인 동반자로 기능한다는 선행연구와 궤를 같이한다[5]. 결국 AI의 지원을 받으면서 어떻게 학습자의 주도적 노력을 이끌어내고 이를 공정하게 평가할 것인가가 프로그래밍 교육의 성패를 결정짓는 과제가 될 것이다.

5.2 기초개념 학습과 프로젝트 수행의 유기적 연계

생성형 AI 시대에는 AI가 생성한 코드의 허점을 식별하고 검증할 수 있는 탄탄한 배경지식이 필수적이다. 그러나 전통적인 주입식 강의만으로는 AI에 익숙해진 학생들의 능동적인 학습 동기를 유발하기에 한계가 있다. 대안으로 주목받는 전면적 프로젝트 기반 학습(PBL) 역시 교과목 위계와 기초 개념을 증시하는 국내 대학 환경에 그대로 적용하기에는 실현 가능성이 낮다. 특히 객체지향 등 핵심 설계 원리가 내면화되지 않은 상태에서 AI를 활용할 경우, 구조적 일관성이 결여된 파편화된 코드를 양산할 위험이 크다.

따라서 본 연구는 기초 개념 강의와 단계별 연습 과제를 통해 충분한 훈련을 거친 후 프로젝트를 도입하는 계층적 교수 설계를 적용하였다. 학생들은 소규모 과제를 통해 설계 기법을 미리 습득함으로써, 대규모 프로젝트 단계에서 AI를 단순한 코드 생성기가 아닌 대등한 협업 파트너로 활용할 수 있는 안목을 갖추게 되었다. 결론적으로 AI 시대의 프로그래밍 커리큘럼은 기초를 생략하는 것이 아니라 더욱 견고히 다지는 방향으로 강화되어야 한다. 이러한 단계적 접근(Scaffolding Approach)은 학습자들이 AI에 매몰되지 않고 기술을 능동적으로 제어하며 실제 시스템을 구현하는 역량을 갖추는 데 기여할 것이다.

5.3 프로젝트 참여도 확대: AI 효과의 고무적 지표

본 연구에서 주목한 가장 고무적인 결과는 생성형 AI를 본격적으로 도입한 2025년에 팀 프로젝트의 협업 구조가 전년 대비 안정화 및 상향 평준화되었다는 점이다. 정량적으로 팀별 기여도 분산의 평균은 감소하였고 동시에 표준편차(SD)가 0.1169에서 0.0844로 약 27.8% 감소한 결과를 얻었다. 2024년은 팀별 협업의 정도가 편차가 큰 편이었으나, 2025년의 수치는 팀이 고른 참여 양상을 보이게 되었음을 입증한다.

이는 과거 팀 프로젝트의 고질적인 병폐였던 상위권 학생들의 독주와 하위권 학생들의 소외 현상이 상당부분 완화되었음을 시사한다. 구체적으로 AI는 기술적 역량이 부족한 학생들에게 포기하지 않고 과업을 시도할 수 있는 심리적 안정감과 실행력을 제공한 것으로 보인다. 이는 단순히 코드 생성에 그치지 않고, 설계 및 역할 분담, 코드 통합 단계에서도 효과적으로 작용하여 최종 산출물의 품질과 수준을 상향 평준화하는 데 기여하였다.

특히 2024년 사례에서 리더급 학생들이 다른 팀원들이 코드 재작성과 통합에 극심한 부담을 토로했던 것과 대조적으로, 2025년에는 GitHub 등의 협업 도구와 AI의 조력이 결합되어 팀원들의 고른 참여를 이끌어낼 수 있었다. 2025년에도 리더의 기여도 집중은 여전히 있었으나 결과적으로 리더는 시스템의 전체적인 완성도를 높이는 조율자의 역할에 집중할 수 있었으며, 이는 팀 프로젝트 운영의 교육적 가능성을 보여주는 중요한 성과라 할 수 있다.

그러나 본 연구의 긍정적 결과를 오로지 생성형 AI의 독립적인 효과로만 해석하기에는 일정한 한계가 존재한다. 우선 2025년 수업에서는 AI 도입과 병행하여 GitHub 등 협업 도구의 활용도가 전년 대비 증가하였으며, 사회 전반의 관심 고조에 따른 학생들의 전반적인 AI 리터러시 역량 강화가 복합적으로 작용했을 가능성이 크다.

또한, AI 활용의 실질적 효과는 팀 프로젝트의 성격, 팀원 간의 역량 차이, 교수자의 가이드라인 수준 등 개별 교육 환경에 따라 상이하게 나타날 수 있으므로, 본 연구의 결과를 다른 프로그래밍 교육 환경으로 일반화하는 데에는 제약이 따른다. 그럼에도 불구하고, 이러한 외부 환경의 변화가 AI 시대의 실제적인 학습 맥락임을 고려할 때, 본 연구가 포

착한 기여도 분포의 안정화 추세는 기술 생태계의 변화가 교육 현장의 고질적인 협업 불균형을 해소하는데 긍정적 역할을 할 수 있음을 시사한다.

5.4 AI 활용과 평가 패러다임: 자기 규율과 과정 중심의 접근

정답의 유무로 성과를 측정하던 전통적 평가 방식은 AI 활용 환경에서 그 변별력을 상실해가고 있다. 상대평가 위주의 국내 교육 여건상 통제된 환경에서의 지필시험을 당장 폐지하기는 어려우나, AI 활용과 AI 없는 평가 사이의 간극은 해결해야 할 숙제이다.

이러한 문제에 대한 명쾌한 해법은 아직 부재한 것으로 보인다. 그러나 본 연구는 기술적 통제라는 단기적 처방보다는, 학습자 스스로가 AI를 도구로서 올바르게 책임감 있게 활용할 수 있는 자기 규율(Self-Discipline) 역량과 자기 주도 학습 능력을 강화하는 근본적인 교육적 접근이 필요함을 제안한다.

또한 본 연구에서 제시한 기여도 정규화 지표는 프로젝트 기반 수업에서 개별 기여도를 객관적으로 증명하는 유효한 도구가 될 수 있을 것으로 기대된다. 기여도의 평준화라는 긍정적 토대 위에 AI와 상호작용하는 과정을 평가하고 가이드하는 교수자의 역할이 자리잡는다면, 프로그래밍 교육은 기술에 잠식당하지 않고 새로운 진화의 길을 찾을 수 있을 것이다.

5.5 초중고 교육 및 AI 리터러시 함양에 대한 시사점

본 연구에서 확인된 대학 수업의 AI 활용 양상과 기여도 변화는 초중고 공교육 현장의 AI 및 소프트웨어 교육 방향에 다음과 같은 시사점을 제공한다.

첫째, 컴퓨팅 사고력의 본질적 함양과 비판적 수용 능력을 강화해야 한다. AI 결과물을 검증하고 통합하려면 탄탄한 기초 개념이 필수적이다. 따라서 초중고 단계에서는 단순 코딩 학습보다 문제 분해와 추상화 등 기초 역량을 다지는 것이 중요할 것이다.

둘째, AI를 활용하는 경우 열린 문제해결 중심의 PBL 방식을 확대 도입할 필요가 있다. 선행 연구에 따르면 열린 문제해결 과정에서의 AI 도입은 학습자의 고차원적 사고와 메타인지를 제고하는 기제로 작용하며[5], 본 연구의 결과 또한 AI가 팀 프로젝트 내에서 학생들의 실질적인 참여와 기여를 이끌어내는 유효한 도구임을 뒷받침한다. 또한 AI를 통한 자신감 제고와 결과물 도출을 통해 초중고 교육 현장에서 학습 소외를 방지하고 보편적 학습 설계의 기반을 제공할 수 있을 것이다.

셋째, 기술적 통제를 넘어선 자기 규율 중심의 디지털 시민성 교육이 필요하다. AI 접근을 물리적으로 차단하기 어려운 환경에서 학습자 스스로 활용 방법과 정도를 조절하는 역량은 미래의 핵심 경쟁력이다. 이를 위해서는 결과 중심의 평가에서 벗어나 AI와의 협업 과정을 중시하는 평가 패

러다임이 정착할 때, 학생들은 기술에 의존하지 않고 이를 주도적으로 통제하는 능력을 습득할 것이다.

6. 결론

본 연구는 생성형 AI가 도입된 프로그래밍 교육 환경에서 학습자의 인식 변화와 함께 팀 프로젝트 협업 양상에 미치는 영향을 실증적으로 분석하였다.

우선 협업 구조 분석 결과를 이용해 생성형 AI 도입 이후 팀 내 기여도 불균형이 완화되었음을 확인하였다. 동료 상호 평가 기반의 기여도 정규화 값(RCi)을 이용하여, 2024년 대비 2025년에 극단적 저기여자가 감소하고 부분 기여 구간의 비율이 약 12%p 증가하며 실질적인 참여층이 두터워졌음을 확인하였다. 또한 팀별 기여도 분산(Vg)의 평균값은 약 11.4% 감소하였다. 소규모 표본의 특성상 t-검정의 유의수준($p=0.713$)에는 도달하지 못했으나, 분산의 표준편차(SD)가 0.1169에서 0.0844로 약 27.8% 급감한 점은 주목할 만한 변화이다. 이는 AI라는 기술적 발판을 통해 수업 전체의 협업 품질이 상향 평준화되며 안정화되었음을 입증한다.

본 연구는 제한된 표본의 특정 수업 사례 연구로서 일반화에는 한계가 있으나, 생성형 AI가 교육 현장에서 개별 학습자의 고른 참여를 유도하고 협업의 질을 균등하게 관리하는 데 기여할 수 있음을 정량적으로 보여주었다는 데 의의가 있다.

이러한 정량적 변화는 학습자들의 심리적 수용성과 교육적 효능감에 의해서도 뒷받침된다. 추가적인 인식조사 결과, AI 활용 경험이 부족했던 학생들도 단기간에 높은 자기 효능감을 보이며 AI를 필수 학습 도구로 수용하였으며, AI는 단순한 코드 생성기를 넘어 프로젝트에서 학습자가 부딪히는 난관을 극복하도록 돕는 발판 역할을 수행하며 학습 효과 향상에 기여하고 있음이 나타났다.

이러한 분석 결과를 종합하여 본 연구는 AI 시대 프로그래밍 교육을 위한 세 가지 방향을 제언한다. 1) 기초 개념 습득 후 프로젝트를 수행하는 단계적 학습 위계 정립, 2) 기여도 등 지표를 활용한 과정 중심 평가 체제 도입, 마지막으로 3) 학습자 스스로 AI 활용 방식과 정도를 조절하는 자기 규율 역량 함양이 필요하다.

본 연구의 한계를 보완하기 위해 향후 연구에서는 다양한 수준의 학습자 집단을 대상으로 생성형 AI의 효용성을 검증하고, AI 활용 능력이 팀 내 기여도에 미치는 상관관계를 보다 정밀하게 분석할 필요가 있다. 특히 동료 상호 평가가 지닌 주관성을 보완하기 위해 협업 과정에서 발생하는 GitHub의 로그 자료 등을 활용한 정량적 분석을 병행함으로써, 학습자들의 실제 협업 양상을 더욱 입체적으로 규명하는 후속 연구가 요구된다. 나아가 이러한 다각적 분석을 토대로 국내 교육 환경에 맞는 팀 프로젝트 평가 모델이 도출될 수 있기를 기대하며, 본 연구가 AI 시대 프로그래밍 교육의 새로운 교수학습 패러다임을 정립하는 기초 자료가 되

기를 기대한다.

참고문헌

- [1] Denny, P., et al. (2024). Computing Education in the Era of Generative AI. *Communications of the ACM* 67(2), 56-67. <https://doi.org/10.1145/3624720>
- [2] Franklin, D., et al. (2025). *Generative AI in Computer Science Education: Challenges and Opportunities*. Cambridge University Press. <https://doi.org/10.1017/9781009581738>
- [3] Shin, S., Yun, B., Chin, H., & Cho, Y. (2025). The Impact of Generative AI on the Future of Software Developer Hiring: Insights from an Expert Delphi Survey. *Journal of Korean Association for Regional Information Society*, 28(2), 69-110. <https://doi.org/10.22896/karis.2025.28.2.003>
- [4] adimalla, Y., Maher, L., Najjar, N., & Simon, C. (2025). The Impact of AI Tools on the Student Programming Experience: Interactions, Attitudes & Perceptions. In *2025 IEEE Frontiers in Education Conference (FIE)*, 1-10. <https://doi.org/10.1109/FIE63693.2025.11328137>
- [5] Wang, J. & Fan, W. (2025). The effect of ChatGPT on students' learning performance, learning perception, and higher-order thinking: insights from a meta-analysis. *Humanit Soc Sci Commun* 12(1), 1-12. <https://doi.org/10.1057/s41599-025-04787-y>
- [6] Yilmaz, R. & Yilmaz, K. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, 4, Article 100147. <https://doi.org/10.1016/j.caeai.2023.100147>
- [7] Alanazi, M., Soh, B., Samra, H. & Li, A. (2025). The influence of artificial intelligence tools on learning outcomes in computer programming: A systematic review and meta-analysis. *Computers*, 14(5), 185. <https://doi.org/10.3390/computers14050185>
- [8] Jang, E. & Oh, K. (2025). Analysis of the Impact of Generative AI Utilization on Computational Thinking and Problem-Solving Ability. *The Journal of Korean Association of Computer Education*, 28(7), 83-98.
- [9] Lepp, M. & Kaimre, J. (2025). Does generative AI help in learning programming: Students' perceptions, reported use and relation to performance. *Computers in Human Behavior Reports*, 18, Article 100642. <https://doi.org/10.1016/j.chbr.2025.100642>
- [10] Rahe, C. & Maalej, W. (2025). How do programming students use generative AI? *Proceedings of the ACM on Software Engineering*, 2(FSE), 978-1000. <https://doi.org/10.1145/3715762>
- [11] Li, H. & Ma, B. (2025, December). CodeRunner Agent: Integrating AI Feedback and Self-Regulated Learning to Support Programming Education. *Proceedings of International Conference on Computers in Education*. <https://library.apsce.net/index.php/ICCE/article/view/5975>
- [12] Park, J. (2025). Teaching Programming with AI Tools - A Case Study for Non-Technical People. *Journal of Practical Engineering Education*, 17(1), 31-40.
- [13] Ada, M. (2026). Programming Challenges and Perceptions: A Study of Separate Groups before and after the Release of ChatGPT, *ACM Transactions on Computing Education*, 26(2), 1-29. <https://doi.org/10.1145/3777904>
- [14] Hwang, H. & Lee, J. (2025). Designing a Programming Course Model Using AI-Based Pair Programming Techniques. *The Journal of Korean Association of Computer Education*, 28(2), 13-22.
- [15] Kim, S. (2023). Developing Code Generation Prompts for Programming Education with Generative AI. *The Journal of Korean Association of Computer Education*, 26(5), 107-117. <https://doi.org/10.32431/kace.2023.26.5.009>
- [16] Guo, N., Han, B., & Qiao, J. (2025). Exploration of the Reform of Comprehensive Practice Course in Advanced Programming Driven by Large Models. *Frontiers in Educational Innovation and Research*, 1(3), 81-91. <https://doi.org/10.62762/FEIR.2025.722615>
- [17] Noh, J. & Chung, H. (2023). The Cognitive and Affective Effects of Project-based SW Class: Focused on Non-Majors. *The Journal of Korean Association of Computer Education*, 27(3), 345-359.
- [18] Hwang, S. (2025). Practice-Oriented Software Engineering Education Model Utilizing Generative AI. *The Journal of Korean Association of Computer Education*, 28(3), 65-77.
- [19] Dawson, G., Deer, R., & Boguslawski, S. (2025). Cognitive dissonance in programming education: A qualitative exploration of the impact of generative AI on application-directed learning. *Computers in Human Behavior Reports*, 19, 100724. <https://doi.org/10.1016/j.chbr.2025.100724>
- [20] Harrison, J. L., Gorman, J. C., Reitman, J. G., & D'mello, S. (2023). Toward the validation of a novel measure of individuals' influence during team collaborations. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 67(1), 290-295. <https://doi.org/10.1016/j.chbr.2025.100724>

org/10.1177/21695067231192643

- [21] obko, P. & Roth, P. L. (2004). The four-fifths rule for assessing adverse impact: An arithmetic, intuitive, and logical analysis of the rule and implications for future research and practice. *Research in Personnel and Human Resources Management*, 23, 177–198. [https://doi.org/10.1016/S0742-7301\(04\)23004-3](https://doi.org/10.1016/S0742-7301(04)23004-3)
- [22] Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J. & Santos, E. A. (2023). Programming is hard-or at least it used to be: Educational opportunities and challenges of ai code generation. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, 1, 500-506. <https://doi.org/10.1145/3545945.3569759>



이은정

· 1988년 서울대학교 계산통계학과(이학사)
· 1990년 한국과학기술원 전산학과(전산학석사)
· 1994년 한국과학기술원 전산학과(공학박사)
· 2001년~현재 경기대학교 AI컴퓨터공학부 교수

✚ 관심분야 : 프로그래밍언어, 컴퓨터교육, AI활용
교육방법론

✉ ejlee@kyonggi.ac.kr

부록

본문의 영문 초록 부분은 제미니 3 버전을 이용하여 생성한 후 저자가 수정하여 완성하였으며, 참고문헌 요약을 위해 AI 요약 기능을 활용하였음.

〈표 1〉 강좌의 수업 운영 및 평가 체계

구분	주요 내용
개요	객체지향 프로그래밍 (자바 기반)
대상 및 수강생	컴퓨터공학 전공 2학년, 150명 (3개반)
수업 방법	플립 러닝 (온라인사전학습: 개념과 이론, 대면 수업: 실습)
과제 (10%)	프로그래밍 과제 (개별, 5회)
시험 (55%)	중간/기말 필기 시험 (50%), 실기 시험 (5%)
팀프로젝트 (10%)	6주 팀 프로젝트 (개별 전 과정)

〈표 2〉 수업에서 생성형 AI 사용 가이드 (2025)

구분	AI 활용 정책
학습	프롬프트 작성 연습, 학습 발판으로서의 AI
과제	AI 활용 정책 프롬프트 작성 연습, 학습 발판으로서의 AI
프로젝트	전과정 AI 사용, 결과 보고
평가/테스트	필기 및 실기 시험 : AI 사용 금지

〈표 3〉 팀프로젝트 설계 및 운영 개요

구분	요구사항 및 평가
기술적 요구사항	OOP 원칙 준수, 파일 기반 입출력 활용, 스윙 기반 GUI 개발
수행 과정	중간 점검 및 피드백, 최종 발표 및 시연, 결과 보고서
평가 기준	팀: 완성도, 협업 개별: 동료 상호평가, 개별보고서

〈표 4〉 설문조사 구성과 조사 시기

구분	주요 지표 (대표 예시)	T1	T2	T3
프로그래밍 자신감	자바 및 OOP 개념에 대한 자신감	●	●	
AI 활용 역량	생성형 AI 사용 능력 및 자신감	●	●	●
AI 효능감	학습, 코딩, 디버깅에서 AI의 효능감	●	●	●
AI 활용 배경	사전 코딩 역량 및 AI 사용 빈도	●		
학습 태도	프로그래밍 접근 방식 및 문제해결 태도	●		
AI 위험	AI 의존성과 공정성에 대한 우려		●	●
팀프로젝트 경험	협업과 프로젝트 개발에 대한 AI의 도움 정도			●
응답자 수		142	67	33

〈표 5〉 데이터 수집 및 분석 개요

구분	2024	2025	비고
전체 학생	128	152	3개반
전체 팀	24	23	팀평가 기준: 완성도, 협업
분석대상 팀	15	15	상중하 성취도에 따른 분포
분석대상 보고서	84	107	개별 프로젝트 보고서: 분석 대상 팀의 모든 학생
평균 팀규모	5.6	7.1	

〈표 6〉 팀기여도 지표 계산 과정 예시

No	단계	결과 예시																																				
1	점수 수집	기여도 점수 (팀내 동료 평가)																																				
2	팀 기여도 행렬 $N \times (N-1)$ 행렬	<table><tr><td>0</td><td>30</td><td>25</td><td>15</td><td>17</td><td>13</td></tr><tr><td>20</td><td>0</td><td>35</td><td>15</td><td>15</td><td>15</td></tr><tr><td>20</td><td>35</td><td>0</td><td>15</td><td>16</td><td>14</td></tr><tr><td>18</td><td>30</td><td>25</td><td>0</td><td>15</td><td>12</td></tr><tr><td>17</td><td>32</td><td>23</td><td>15</td><td>0</td><td>13</td></tr><tr><td>15</td><td>35</td><td>25</td><td>14</td><td>11</td><td>0</td></tr></table>	0	30	25	15	17	13	20	0	35	15	15	15	20	35	0	15	16	14	18	30	25	0	15	12	17	32	23	15	0	13	15	35	25	14	11	0
0	30	25	15	17	13																																	
20	0	35	15	15	15																																	
20	35	0	15	16	14																																	
18	30	25	0	15	12																																	
17	32	23	15	0	13																																	
15	35	25	14	11	0																																	
3	1개인 기여도 평균 C_i	<table><tr><td>18.0</td><td>32.4</td><td>26.6</td><td>14.8</td><td>14.8</td><td>13.4</td></tr></table>	18.0	32.4	26.6	14.8	14.8	13.4																														
18.0	32.4	26.6	14.8	14.8	13.4																																	
4	정규화된 개별 기여도 RC_i	<table><tr><td>-0.10</td><td>0.62</td><td>0.33</td><td>-0.26</td><td>-0.26</td><td>-0.33</td></tr></table>	-0.10	0.62	0.33	-0.26	-0.26	-0.33																														
-0.10	0.62	0.33	-0.26	-0.26	-0.33																																	
5	팀 기여도 분산 V_g	$V_g = 0.1246$																																				

〈표 7〉 학습 유형에 따른 생성형 AI 사용 인식 비교

구분	개인 과제 및 시험	팀프로젝트
내용	지식습득 및 평가 (정해진 답이 있는 문제)	복잡한 문제 해결 (개방형 문제)
주요 태도	유보적, 주의적	적극적, 수용적
긍정 응답율	사용 도움 71.6%	효능감 89.4%
부정 / 우려	기초과정에서는 사용 안해야 66.7%	AI 없이 수행 어렵다 63.6%
주요 도움 분야	디버깅 (51%), 개념 이해 (15%)	문제 해결 (30%), 전과정 (37.5%)
주요 우려 사항	의존성 증대, 스스로 사고 능력 저하	

〈표 8〉 개인 기여도 정규화값(RC_i)의 구간 분포

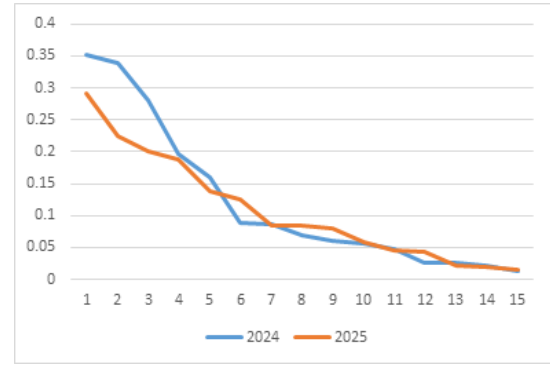
기여도 구간	해석	2024 (%)	2025 (%)	Change(%p)
$0.5 \leq RC_i$	핵심 리더	7.1	6.5	-0.6 ↓
$0 \leq RC_i < 0.5$	평균이상 기여자	42.9	36.4	-6.5 ↓
$-0.5 < RC_i < 0$	부분기여자	40.4	52.3	11.9 ↑
$RC_i \leq -0.5$	극단적저기여자 (무임승차)	9.5	4.7	-4.8 ↓

〈표 9〉 기여도 데이터 분석 요

2024 (AI 미도입)	2024 (AI 미도입)	2025 (AI 도입)	변화량
팀내 기여도 분산 (V_s) 평균	0.1217	0.1078	11.4% ↓
최대 분산 팀	0.353	0.292	17.2% ↓
V_s 의 표준편차 (SD)	0.1169	0.0844	27.8% ↓
개별기여도 저기여자 구간 비율 ($-0.2 \leq RC_i < 0$)	21.4%	30.8%	11.9%p ↑

· N = 15

· p = 0.713, CI = 95%, Cohen's d = 0.136



〈그림 1〉 팀 기여도 분산 비교 (연도별 내림차순) 부록 〈표 18〉 참조

〈표 10〉 학기초 수강생 설문

No	학기초 설문 문항
1	배열과 반복문의 동작 원리를 설명할 수 있다.
2	클래스로 객체를 만들고 메소드를 설계할 수 있다
3	여러 개의 클래스를 연관지어 간단한 프로그램을 구성할 수 있다
4	오류 메시지를 읽고 스스로 디버깅할 수 있다.
5	자바로 과제를 스스로 완성할 자신감이 있다.
6	AI에게 구체적 프롬프트를 작성하는 법을 알고 있다.
7	AI가 준 코드를 수정·통합할 자신감이 있다
8	자바 실력 자기 평가(매우우수...매우부족)
9	AI 코딩도구 사용 빈도(최근 3개월)
10	주로 사용하는 도구(예: ChatGPT, Copilot 등)

〈표 11〉 학기중 설문 (중간고사 직후)

No	학기 중 설문 문항
1	새로운 것을 배우기 좋아한다
2	과제할 때 시간가는 줄 모르고 한다
3	여러 가지 데이터로 테스트하는것을 즐긴다
4	높은 수준의 기능을 사용해 보고 싶다
5	오류가 나면 계속하고 싶지 않다
6	새로운 기능을 적용하려고 노력한다
7	다른 학생보다 더 좋은 기능과 코드를 만들고 싶다
8	코딩할 때 남들보다 더 오래 걸린다
9	코딩을 한번 시작하면 끝까지 하는 편이다.
10	코딩할 때 AI의 도움 받기를 좋아한다
11	AI의 개념과 방법 설명이 도움이 많이 된다.
12	AI가 생성해준 코드를 이해하고 수정하는 것이 처음부터 혼자 코딩하는 것보다 학습과 과제 완성에 도움이 된다
13	AI를 이용해 과제의 요건이나 요구사항에 맞게 코드를 생성하는 것이 어렵다
14	이 과목에서 AI 사용 프롬프트 교육이 유익하고 필요하다고 생각한다
15	AI 사용의 문제점(의존성, 부작용)으로 어느 것이 가장 심각하다고 생각하나요? (단일선택) 1) 코딩을 하려는 동기 저하 2) 코딩에서 발생한 문제를 직접 해결하는 끈기 저하 3) 알고리즘과 문제해결 방법을 스스로 생각하려는 노력 저하 4) 새로운 기능이나 기법을 도입해서 해결해 보려는 노력 저하 5) AI 없이는 문제해결을 시도하지 못하는 문제

〈표 13〉 학기중 설문 조사 결과

No	매우 그렇다	그렇다	보통	아니다	전혀 아니다	긍정 비율	부정 비율
1	26	38	3	0	0	95.52	0
2	21	30	11	5	0	76.12	7.46
3	16	24	22	5	0	79.70	7.46
4	26	32	8	1	0	86.57	1.49
5	2	8	22	25	10	14.95	52.24
6	18	39	8	2	0	85.08	2.99
7	32	31	1	3	0	94.03	4.48
8	18	24	15	10	0	62.69	14.93
9	16	28	18	4	0	65.67	5.97
10	12	36	12	3	4	71.64	10.45
11	39	23	4	0	0	92.54	0
12	12	27	18	7	3	58.21	14.93
13	0	2	12	28	25	2.99	79.10
14	27	29	7	3	1	83.58	5.97
15	2) 25 (37.31%) 3) 9 (13.43%) 5) 32 (47.76%)						

응답자수: 67명

〈표 14〉 기말 설문

No	학기말 설문 문항
1	자바 언어의 새로운 기능과 개념을 배운 것이 자바 능력 향상에 도움이 되었다.
2	시험볼 때 AI의 도움을 받을 수있어야 한다
3	프로그래밍을 배우는 단계에서 AI를 사용하지 않아야 한다
4	프로젝트 수행에서 AI 사용이 도움이 되었나요?
5	프로젝트 수행을 통해 자바 능력이 향상되었다
6	큰 프로그램을 작성하면서 자바 언어 활용 능력이 향상되었다
7	AI 도움이 없었다면 프로젝트 진행이 어려웠을 것이다
8	자바 언어 학습에 자신감이 높아졌다
9	자바와 개발 공부에 AI를 잘 사용할 수 있다
10	프로그래밍 실력이 향상되었다
11	개발에 필요한 것을 AI를 통해서 해결하는데 자신감 생겼다
12	AI 도움으로 프로젝트 개발에 자신감이 생겼다
13	AI가 없으면 코딩이 어렵다
14	AI가 생성해주는 코드의 답변 수준이 점점 좋아지는 것을 느낀다
15	AI가 생성해주는 코드가 매번 달라져서 일관성이 없다
16	AI로 생성한 코드는 결국 내가 다시 이해하고 고쳐야 한다
17	학습할 때 AI의 응원이나 격려가 도움이 된다
18	학습하려는 동기부여를 위해 AI의 지원이 도움이 된다
19	코딩에 대한 두려움을 해소하는데 AI의 격려가 도움이 된다
20	과제수행에서 AI의 도움 (단일응답) 1) 개념 이해와 설명 2) 예제 코드 이해 3) 프로그래밍 오류 해결 4) 다양한 방식의 코드 작성 방법 4) 기본적인 코드 생성 6) 별로 도움이 되지 않았다.
20	프로젝트에서 AI의 도움 (단일응답) 1) 데이터 만들기 2) 기능 설계 및 기획 3) GUI 코드 생성 4) 문제 해결하기 5) 코드 디버깅 6) 모든 면에서 도움이 되었다 7) 별로 도움이 되지 않았다

〈표 16〉 2024년 팀프로젝트 팀별 기여도 정규화 행렬

No	정규화된 개인별 기여도 평균(2024)								팀분산
1	0.08	0.08	0.03	-0.31	0.02	-0.21			0.0261
2	-0.51	0.13	0.52	-0.69	0.25	0.31			0.1961
3	-0.27	0.52	-0.3	-0.05	0.1				0.0892
4	0.06	-0.26	-0.08	0.48	-0.21				0.0695
5	-0.09	-0.53	-0.39	-0.2	1.09	0.12			0.2806
6	-0.22	0.36	0.06	0.18	-0.26	-0.11			0.0479
7	0.16	0.16	-0.5	0.36	-0.01				0.0862
8	-0.61	-0.56	0.71	0.75	-0.04				0.3526
10	0.39	-0.59	0.45	0.04	-0.31				0.1601
11	-0.09	-0.29	0.32	0.17	-0.32	0.18			0.0597
12	-0.2	-0.03	0.28	0.04	-0.18	0.09			0.0269
13	-0.07	-0.03	0.14	0.13	-0.17				0.0139
14	-0.05	-0.1	-0.05	-0.06	-0.06	0.32			0.0208
15	-0.87	0.09	0.08	0.18	-0.48	1			0.3390
16	0.07	0	-0.1	-0.35	0.38				0.0564

N=83, RCI평균=0.0008, RCI분산=0.3514, 팀분산 평균=0.1217

〈표 15〉 학기중 설문 조사 결과

No	매우 그렇다	그렇다	보통	아니다	전혀 아니다	평균	분산
1	15	14	2	1	1	87.87	6.06
2	2	2	7	13	9	12.12	66.66
3	3	5	4	15	6	24.24	66.66
4	31	0	2	0	0	93.94	0
5	9	18	5	0	1	84.85	3.03
6	10	18	4	0	1	84.85	3.03
7	10	21	1	1	0	93.94	3.03
8	8	12	12	1	0	60.66	3.03
9	11	17	5	0	0	84.85	0
10	8	18	5	2	0	78.79	6.06
11	10	21	2	0	0	93.67	0
12	10	18	5	0	0	84.85	0
13	3	18	7	5	0	63.64	15.15
14	9	16	7	1	0	75.75	3.03
15	2	4	13	11	0	18.18	33.33
16	10	13	9	1	0	69.69	3.03
17	3	7	7	14	2	30.30	48.48
18	4	22	5	2	0	78.78	6.06
19	3	19	9	2	0	66.67	6.06
20	1) 6(19.4%) 2) 5 (16.1%) 3) 16 (51.6%) 4) 3 (9.7%) 5) 0 6) 1 (3.2%) (응답 31)						
21	1) 3(9.4%) 2) 1 (3.1%) 3) 6 (18.8%) 4) 9 (28.1%) 5) 1 (3.1%) 6) 12 (37.5%) (응답 32)						

응답자수: 33명

〈표 17〉 2025년 팀프로젝트 팀별 기여도 정규화 행렬

No	정규화된 개인별 기여도 평균(2024)								팀분산
1	0.69	0.27	0.03	-0.09	-0.2	-0.3	-0.52		0.1375
2	0.46	0.14	0.1	-0.01	-0.03	-0.72			0.1246
3	1.18	0.25	0.08	-0.09	-0.17	-0.33	-0.66		0.2918
4	0.26	0.1	0.02	-0.12	-0.12	-0.14			0.0213
5	0.49	0.26	0.11	-0.14	-0.19	-0.34			0.0835
6	0.23	0.16	0.01	-0.20	-0.22	-0.26	-0.29		0.0451
7	0.38	0.16	-0.05	-0.13	-0.15	-0.21			0.0427
8	0.41	0.39	0.31	0.26	-0.14	-0.46	-0.78		0.1887
10	0.21	0.06	0.05	0.05	-0.16	-0.21			0.0204
11	0.5	0.26	0.11	-0.14	-0.19	-0.34			0.0835
12	0.54	0.18	0.17	-0.07	-0.13	-0.22	-0.23	-0.42	0.0806
13	0.16	0.14	0.06	0.00	-0.06	-0.12	-0.13	-0.21	0.0157
14	1.21	0.26	0.12	-0.04	-0.24	-0.3	-0.31	-0.33	0.2236
15	0.65	0.04	-0.06	-0.09	-0.09	-0.1	-0.14	-0.15	0.0583
16	1.07	0.2	0.13	-0.01	-0.03	-0.05	-0.41	-0.43	0.2001

N=107, RCI평균=-0.0053, RCI분산=0.3371, 팀분산 평균=0.1078

〈표 18〉 연도별 팀 프로젝트 기여도 분산 비교

년도	각 연도별 내림차순 정렬 및 동일순위 증감을 계산 결과 (본문 Fig. 1)														
2024	0.353	0.339	0.281	0.196	0.16	0.089	0.086	0.07	0.06	0.056	0.048	0.027	0.026	0.021	0.014
2025	0.292	0.224	0.2	0.189	0.138	0.125	0.084	0.084	0.081	0.058	0.045	0.043	0.021	0.02	0.016
증감율	-0.172	-0.34	-0.287	-0.038	-0.141	0.397	-0.031	0.201	0.35	0.034	-0.058	0.587	-0.184	-0.019	0.129