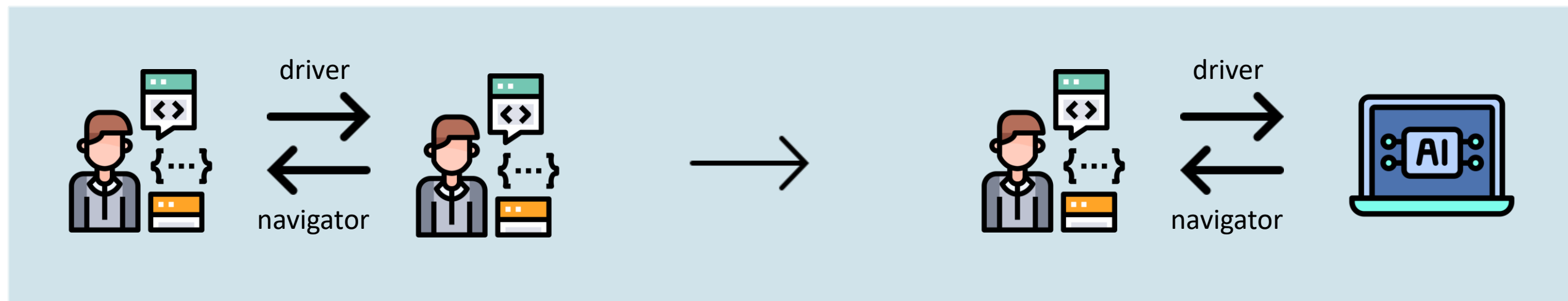


생성 AI 접목 프로그래밍 수업 (Python 기반 인공지능 강의)

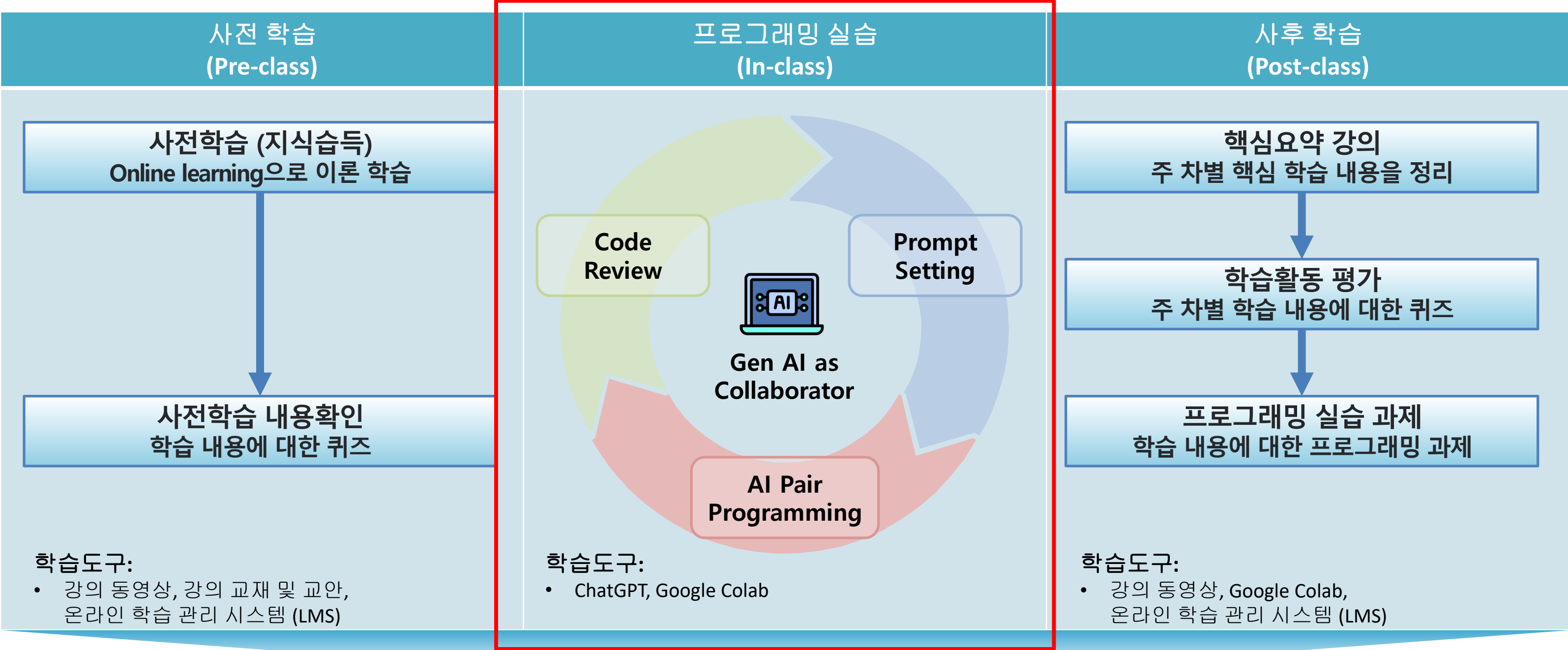
수업모델_프로그래밍 수업 (Python Programming)

- 인공지능 짝 프로그래밍 (AI Pair Programming) 기반 프로그래밍 수업 모형
- 짝 프로그래밍은 두 명의 프로그래머가 한 컴퓨터에서 함께 작업하는 소프트웨어 개발 방법으로, 주기적으로 서로 역할을 바꿔가면서 한 사람은 운전자(driver) 역할을 맡아 주도적으로 프로그램을 작성하고, 다른 한 사람은 항해사(navigator) 역할을 맡아 운전사가 입력하는 코드를 살펴보면서 문제점을 발견하고 대안을 제시하는 프로그래밍 개발 모형
- 본 연구에서는 컴퓨터 프로그래밍 교육 모델(강환수 외, 2020)과 짝 프로그래밍 활용 교육 모델(정충교 외, 2017)을 참고하여 기존의 학생-학생 짝 프로그래밍 구조에서 AI-학생으로 변형하여 수업 모델을 설계



수업모델_프로그래밍 수업 (Python Programming)

- 인공지능 짝 프로그래밍 (AI Pair Programming) 기반 프로그래밍 수업 모형



Outcome

컴퓨팅적 사고 능력

프로그래밍 자기 효능감

학습 동기

ChatGPT기반 AI Pair Programming

수업설계목표

“인공지능 짝 프로그래밍 기법을 이용하여 프로그래밍 학습의 문법 장벽을 줄이고 이론 및 알고리즘에 집중할 수 있게 수업을 설계하여 프로그래밍 실습 강의를 진행”

차시 별 프로그래밍 실습 수업 진행 단계 (상세)

- Step 1: 이론 강의 Review (e.g., Understanding Attention Mechanism) (15 minutes)
- Step 2: Setting Up the Environment (5 minutes) – ChatGPT, GoogleColab, etc.
- Step 3: ChatGPT as the Driver: Hands-On Implementation with ChatGPT (25 minutes)
- Step 4: ChatGPT as the Navigator: Code Review with ChatGPT (25 minutes)
- Step 5: Q&A and Wrap-Up (5 minutes)

• 프로그래밍 실습 강의 예시 (인공지능 12주차 실습)

- **ChatGPT as the Driver:** AI Pair Programming을 위한 Prompt Setting

0. Setting Up the Environment and the Context

- Let's open Google Colab, ChatGPT, and set up the context!
- Prompt:
 - You are an experienced AI Engineer. You and I will be pair programming on a project. We will use the driver-navigator model. I will be the navigator first, then we will switch, then we will switch back again. Do you understand?



Yes, I understand. In the driver-navigator model, the navigator guides the driver, who writes the code. We will alternate roles as specified. Please let me know what the project is about and what the first task is.



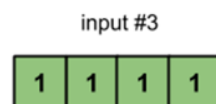
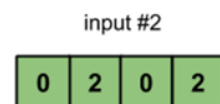
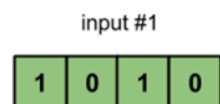
• 프로그래밍 실습 강의 예시 (인공지능 12주차 실습)

- **ChatGPT as the Driver:** 이론에서 배운 내용에 따라 순차적으로 Prompt를 통해 학습 내용을 구현

1. Prepare inputs

- Prompt:

- I'd like you to develop self-attention code using Pytorch step by step. First, prepare 3 inputs, each with dimension 4 as [1, 0, 1, 0], [0, 2, 0, 2], and [1, 1, 1, 1].



Got it! Let's start by preparing our 3 inputs, each with a dimension of 4, using PyTorch.

Here is the code to create these inputs:

```
python
import torch

# Define the inputs
inputs = torch.tensor([
    [1, 0, 1, 0],
    [0, 2, 0, 2],
    [1, 1, 1, 1]
], dtype=torch.float32)

print(inputs)
```

코드 복사

This code sets up our three inputs as a PyTorch tensor. Next, please guide me on the next step.



• 프로그래밍 실습 강의 예시 (인공지능 12주차 실습)

- **ChatGPT as the Driver:** 이론에서 배운 내용에 따라 순차적으로 Prompt를 통해 학습 내용을 구현 - 구현 내용 확인

The screenshot shows a Jupyter Notebook titled "SKKU_AI_2024_Attention.ipynb" with a star icon. The interface includes a top bar with icons for chat, share, settings, and a user profile. Below the title bar, there are tabs for "코드" (Code) and "텍스트" (Text). The main content area displays the following text:

Step 0:

You are an experienced AI Engineer. You and I will be pair programming on a project. We will use the driver-navigator model. I will be the navigator first, then we will switch, then we will switch back again. Do you understand?

Step 1: Prepare inputs

I'd like you to develop self-attention code using Pytorch step by step. First, prepare 3 inputs, each with dimension 4 as [1, 0, 1, 0], [0, 2, 0, 2], and [1, 1, 1, 1].

```
[ ] # @title
import torch

# Define the inputs
inputs = torch.tensor([
    [1, 0, 1, 0],
    [0, 2, 0, 2],
    [1, 1, 1, 1]
], dtype=torch.float32)
```


• 프로그래밍 실습 강의 예시 (인공지능 12주차 실습)

- **ChatGPT as the Navigator:** ChatGPT와 학생 간의 역할 변경을 위한 Prompt 예시

0. Let's inform ChatGPT that we are going to switch roles!

- Let's inform ChatGPT that we are going to switch roles! Then, we will give ChatGPT a description of the intended goal.

- Prompt:

- Great work! With the self-attention code prepared, we will swap roles. I will take the driver's seat, and you will navigate. Please refrain from providing actual code, but feel free to provide pseudocode. Initially, we'll construct the attention layer for image classification.

• 프로그래밍 실습 강의 예시 (인공지능 12주차 실습)

- **ChatGPT as the Navigator** : 이론에서 배운 내용에 따라 직접 구현하고 ChatGPT를 이용하여 Code Review 진행

ChatGPT

Great! Could you provide me with pseudocode for the self-attention layer?

Of course! Here's a pseudocode outline for the self-attention layer:

```
plaintext
function SelfAttentionLayer(input_tensor):
    # Define parameters
    W_key, W_query, W_value = initialize_parameters()

    # Linear transformations
    keys = linear_transform(input_tensor, W_key)
    queries = linear_transform(input_tensor, W_query)
    values = linear_transform(input_tensor, W_value)

    # Calculate attention scores
    attention_scores = dot_product_attention(queries, keys)
```

메시지 ChatGPT

ChatGPT는 실수를 할 수 있습니다. 중요한 정보를 확인하세요.

• 프로그래밍 실습 강의 예시 (인공지능 12주차 실습)

- **ChatGPT as the Navigator** : 이론에서 배운 내용에 따라 직접 구현하고 ChatGPT를 이용하여 Code Review 진행

ChatGPT

```
torch.sqrt(torch.tensor(self.feature_size, dtype=torch.float32))

# Apply softmax
attention_weights = F.softmax(scores, dim=-1)

# Multiply weights with values
output = torch.matmul(attention_weights, values)

return output, attention_weights
...

Is this correct?
```



Yes, the provided code defines a self-attention layer correctly using PyTorch. Let's break down the key components:

1. **Initialization:** The `__init__` method initializes the self-attention layer with the specified `feature_size`. It defines three linear transformations (`nn.Linear`) for the key, query, and value vectors, each transforming input feature vectors of size `feature_size` to output feature vectors of the same size.



메시지 ChatGPT



• 연구결과 (1)

- 연구 대상 : 2024학년도 1학기 인공지능 강의 수강생 19명 (19/30)
- 설문 문항 : 총 55개 문항
 - 컴퓨팅적 사고능력
 - 프로그래밍 자기효능감
 - 학습동기

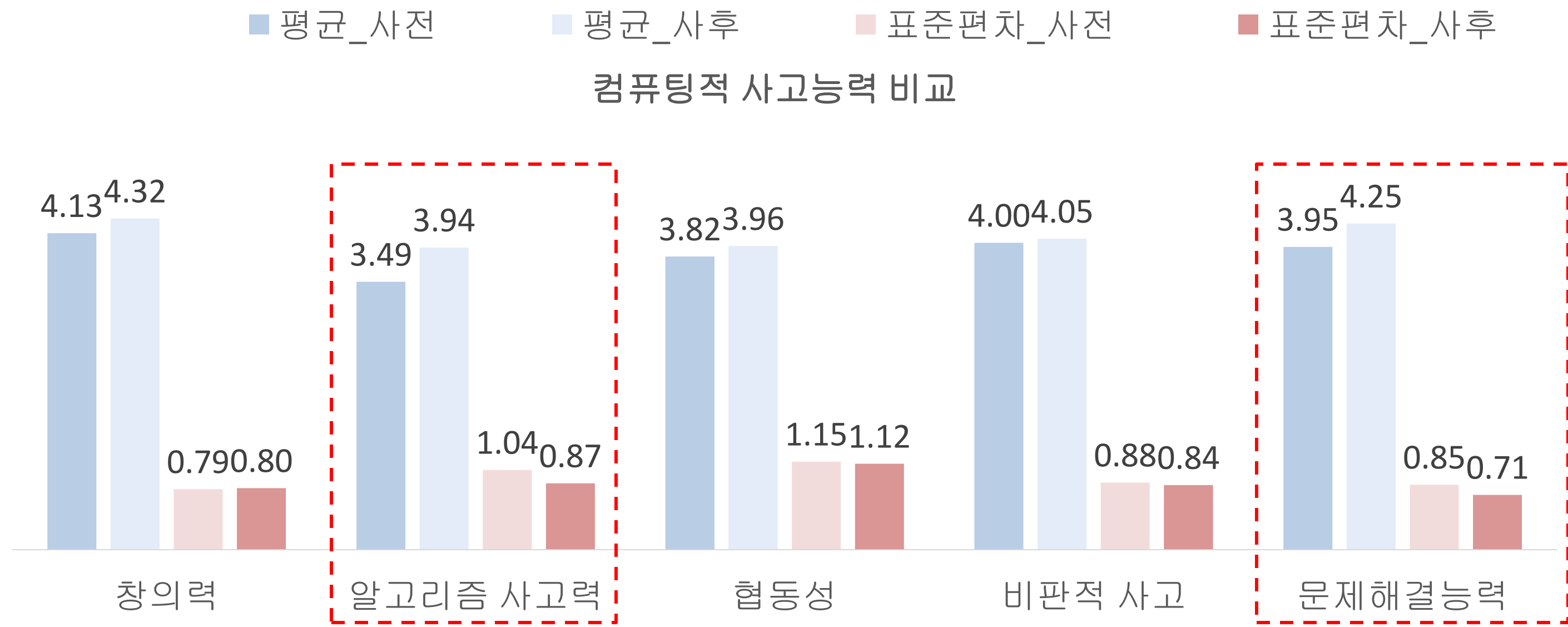


• 연구결과 (2)

□ 주요 결과 :

1. 컴퓨팅적 사고능력 (5점 만점) – 총 25개 문항

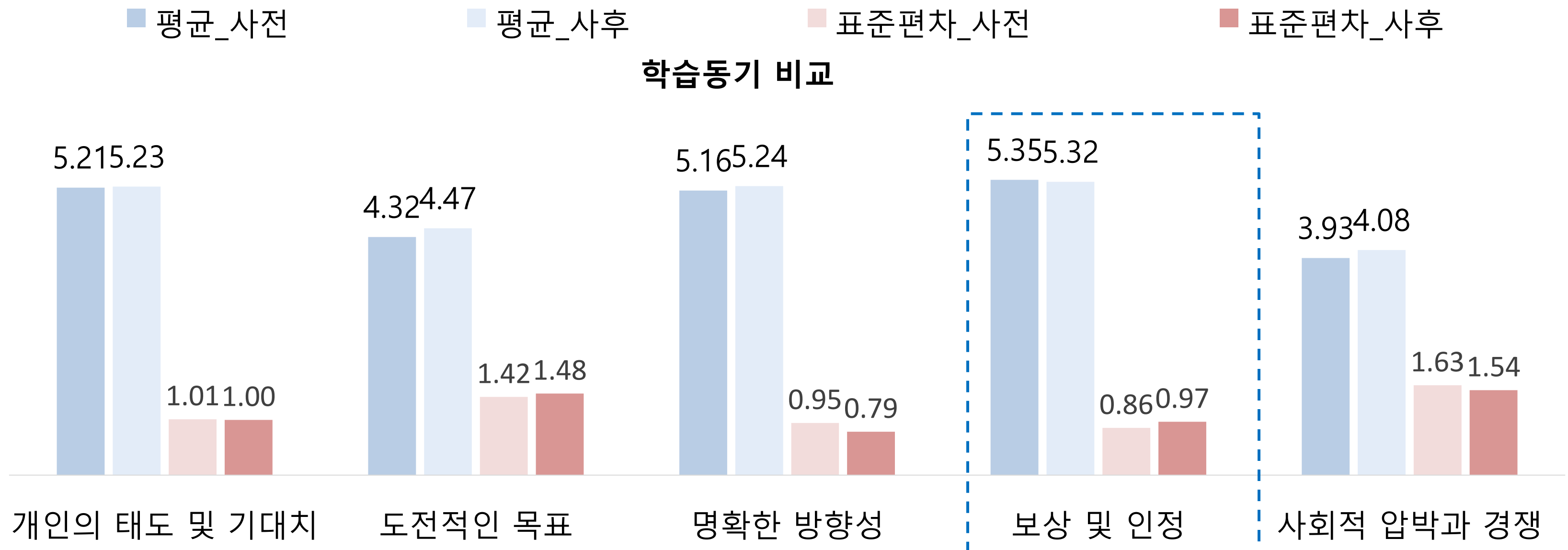
- 모든 항목에서 평균이 **증가**. 특히, 알고리즘 사고력(3.49 → 3.94)과 문제해결능력(3.95 → 4.25)에서 큰 증가를 보임
- 또한 대부분 항목의 표준편차가 감소하여, **참자가들의 능력이 좀 더 균일해진 것을 확인**



• 연구결과 (3)

2. 학습동기 (6점 만점) – 총 15개 문항

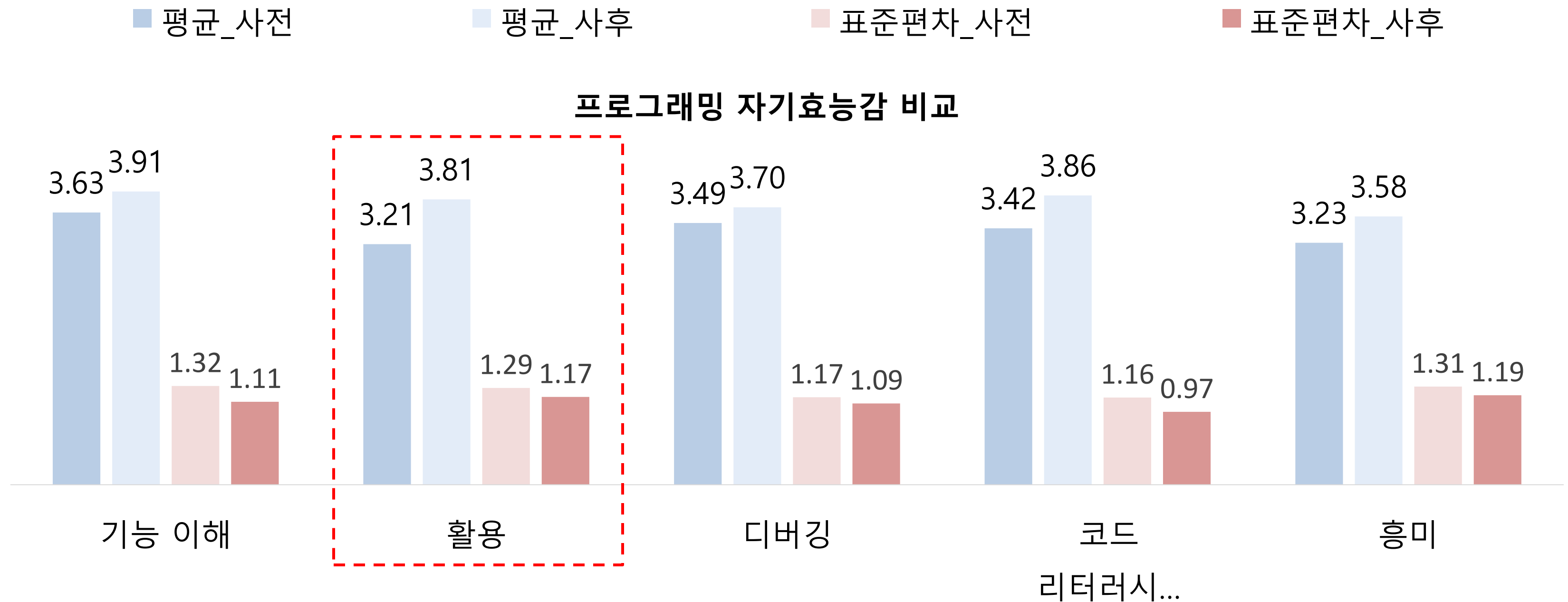
- 대부분의 항목에서 평균이 소폭 **증가** 하였고, '**보상 및 인정**' 항목에 대해서는 소폭 **감소**
- 프로그램 자체가 외부적 보상이나 인정을 크게 강조하지 않는 방식으로 진행되어 오히려 감소한 것으로 보임
- 설문 문항 예시: '강사의 격려와 좋은 코멘트가 학습에 대한 동기를 부여한다.'
- ※ '보상 및 인정' 항목 관련 설문문항에 대한 상세 내용은 슬라이드 39-page에서 확인 가능



• 연구결과 (4)

3. 프로그래밍 자기 효능감 (5점 만점) – 총 15개 문항

- 모든 항목에서 평균이 **증가**하였고, 특히 '활용' 부분에서 가장 큰 증가를 보였음 (3.21 → 3.81)



• 연구결과 (심층 인터뷰 결과 분석 진행중) (1)

Q1: 생성형 AI 활용 강의 수강 후 학습 방법과 문제 해결 능력 변화?

“

”

새로운 개념을 배울 때 모르는 내용을 구글링 하던 방식에서 생성형 AI를 활용하여 코드 구현을 요청하고 예시 코드와 주석을 보면서 질문하는 방식으로 바뀌었고, ';'이나 들여쓰기 등 **에러를 잡아주어** 효율성이 올라갔음

Q2: 강의 참여에 대한 변화?

“

”

기존 실습 방식은 실습시간에 교수님이 작성하는 코드를 보고만 있거나, 직접 작성할 때 막막한 부분이 있었는데 생성형 AI를 활용하여 질문을 편하게 할 수 있어 **진입 장벽이 낮아져** 동기부여를 받았음

• 연구결과 (심층 인터뷰 결과 분석 진행중) (2)

Q3: 생성형 AI를 사용하면서 어려웠던 점은?

“

”

생성형 AI의 횡수 제한, 전체 코드가 아닌 코드 일부만을 넣기 때문에 생성형 AI가 코드 작성을 완벽하게 도와줄 수 없었음

Q4: 생성형 AI를 활용한 강의에서 개선되었으면 하는 점은?

“

”

프롬프트 작성 능력을 향상시키기 위해, 처음부터 교수의 직접적인 도움 없이 AI와 짝프로그래밍을 해보면 좋을 것 같음

Q5: 전반적인 경험

“

”

짝 프로그래밍을 처음 해보았는데 사람과 하기 힘들었던 활동을 AI와 해볼 수 있어서 좋은 경험이 되었음. 다만 AI가 코드를 대신 다 써주는 경향이 있기 때문에 프롬프트 작성 시 제약 조건을 걸어야 학습면에서 도움이 될 것 같음

요약

- **AI 기반 짝 프로그래밍 모델 효과**
 - 학생들의 컴퓨팅적 사고능력 및 프로그래밍 자기 효능감 향상
- **[개선점] 향상된 능력**
 - 알고리즘 사고력, 문제해결 능력, 프로그래밍 활용 능력
 - 프로그래밍의 문법적인 부분 보다 논리와 알고리즘에 집중하여 학습 가능
- **[보완이 필요한 사항] 학습동기 및 프롬프트 설정 문제**
 - AI 코멘트는 강사의 격려처럼 보상으로 인식되지 않음
 - 따라서, 실습 시 적절한 보상 방법 필요
 - 프롬프트 설정 중요성
 - 부적절한 프롬프트 설정 시 AI가 학생 대신 코드 작성
 - 학생의 프로그래밍 역량 향상 방해 가능
- **수업 모델 개선 필요성**
 - 학생이 코드를 구현하고 AI로 코드 리뷰를 진행하는 경우 세밀한 수업모델 설계 필요