

프로그래밍 교육 기초 학습자를 위한 교육 콘텐츠 생성 GPTs 챗봇 구성

Configuring a GPTs Chatbot to Generate Educational Contents for Programming Novice Learners

김슬기[†]
Seulki Kim[†]

요 약

디지털 시대에 모든 사람이 갖추어야 할 기초 역량으로서의 컴퓨팅 사고력을 길러 주며 AI 교육에서 AI 개념의 이해와 현실 세계 적용에 효과적인 교육 방법으로 프로그래밍 교육의 중요성이 높아지고 있다. 이에 본 연구는 프로그래밍 교육과 관련된 선행 연구를 분석하고 이를 적용하며 대형 언어 모델(LLM)을 활용한 프로그래밍 교육 콘텐츠 생성 챗봇을 구성하여 기초 학습자의 어려움을 해결하고자 하였다. 프로그래밍 교육 콘텐츠 생성 챗봇은 OpenAI의 GPTs 플랫폼과 GPT-4o 모델을 사용하여 자기 주도적 프로그래밍 실습을 위한 콘텐츠 생성 및 적절한 피드백을 제공하고 개별 교수·학습 환경을 구성할 수 있도록 설계되었다. 이에 주요 기능으로 프로그래밍 개념학습, 프로그래밍 연습, 프로그래밍 도움을 중심으로 기능을 구성하였다. 구성된 챗봇에 대한 교육적 적합성과 응답의 정확성을 11명의 컴퓨터 교육 전문가의 검토를 통해 평가하였고, 이를 기반으로 사용자 상호작용 흐름과 콘텐츠 생성 지침을 개선하였다. 최종 적합성을 평가한 결과, 챗봇의 교육적 유효성과 응답의 정확성 모두 기준 이상의 합의를 확인하였으며 완성된 챗봇을 GPTs 플랫폼을 통해 배포하고 구성을 위한 지침을 공개하였다. 프로그래밍 교육을 위한 LLM 기반 교육 도구의 가능성을 보여준 본 연구의 결과가 프로그래밍 기초 학습자의 교육 접근성을 높이고 다양한 예제와 개인화된 학습 리소스를 제공하여 학습자의 컴퓨팅 사고력을 증진하는데 도움을 줄 수 있기를 기대한다.

주제어: 프로그래밍 교육, 기초 학습자, 대형언어 모델, 챗봇, GPTs

ABSTRACT

In the digital age, cultivating computational thinking as a fundamental skill for everyone has become increasingly important, and programming education is recognized as an effective method for understanding AI concepts and applying them to the real world. This study aims to analyze prior research related to programming education and apply it by developing a chatbot that generates programming education content using LLM to address the difficulties faced by novice learners. The programming education content generation chatbot is designed using OpenAI's GPTs platform and the GPT-4o model, enabling the creation of content for self-directed programming practice, providing appropriate feedback, and facilitating individualized teaching and learning environments. The primary functions of the chatbot include learning programming concepts, practicing programming, and offering programming assistance. The educational suitability and accuracy of the chatbot's responses were evaluated by 11 computer education experts, and based on their feedback, the user interaction flow and content generation instructions were improved. The final evaluation confirmed that both the educational effectiveness and response accuracy of the chatbot met the required standards. The completed chatbot was then deployed through the GPTs platform, and instructions for its configuration were made available. The results of this study demonstrate the potential of LLM-based educational tools for programming education to enhance the accessibility of education for novice programming learners and provide diverse examples and personalized learning resources to improve learners' computational thinking skills.

Keywords: Programming Education, Novice Learners, LLM, Chatbot, GPTs

[†]정회원: 초당초등학교 교사

1. 서론

디지털 시대의 모든 사람이 갖추어야 할 기초 역량으로서의 컴퓨팅 사고력이 강조됨에 따라 학생들의 역량을 길러 주기 위한 가장 효과적인 방법으로 프로그래밍 교육의 중요성이 높아지고 있다[1-3]. 최근 강조되고 있는 컴퓨팅 사고력을 기반으로 한 AI 교육에서도 AI의 개념에 대한 이해와 지식을 현실적인 상황에 적용하기 위해서는 프로그래밍 중심의 컴퓨팅 실습이 필수적이다. 이에 프로그래밍에 대한 배경지식은 디지털 기술을 기반으로 문제 해결력과 창의적 산출물 개발, AI 관련 고급 프로그래밍 개념과 알고리즘의 이해에 기반이 되어 그 중요성이 더욱 높아지고 있다[4-6].

프로그래밍 교육은 학습자의 문제 해결력, 논리적 사고력 등 고차적인 사고능력을 기르는데 도움을 줄 수 있다는 관점에서 효과성을 확인할 수 있는 다양한 연구들이 선행되었다. 또한 컴퓨터 과학의 기본적인 이해를 구축하고 고등 교육에서 학생들의 지식수준을 심화할 수 있는 기술적 기반을 마련한다는 측면에서 기본 교육의 성격으로 다루어지고 있다. 따라서 입문 과정의 기초 학습자를 위한 다양한 교육적 접근이 이루어지고 있으며 기초 학습자를 위한 프로그래밍 교육과정은 주로 이론적 지식과 지식의 실제 적용을 혼합하여 컴퓨팅 사고력을 기르는데 초점을 맞추고 있다[7,8].

기초 학습자를 위한 프로그래밍 교육에 관한 연구는 주로 프로그래밍의 어려움을 해소하기 위한 다양한 교육학적 측면을 중심으로 논의되어 왔다. 프로그래밍의 어려움은 주로 학습자가 경험하게 되는 교육 환경과 같은 학습자의 외적 요인과 함께 인지적 부담, 학습 동기 부족 등과 같은 학습자 내적 요인을 중심으로 연구되어 왔다[9,10]. 다양한 연구 결과들이 교육 환경 개선에 도움을 주고 있지만 공통적으로 교육 리소스 생성을 위한 개별 교수자의 한계와 대규모 교육 환경에 따른 문제, 학생과의 능동적인 의사소통 부족 등에서 발생하는 문제들이 해결해야 할 과제로 제시되고 있다[11,12].

최근 AI를 활용한 지능형 튜터링 기술이 발전되고 있으며 개별 학생들에게 적합한 피드백을 제공하여 개인화된 교수·학습 환경을 제공해 주기 위한 연구들이 진행되고 있다[13]. 특히 생성형 AI 중 텍스트를 중심으로 사전 학습된 대형 언어 모델(Large Language Model, LLM)을 활용한 교육 콘텐츠의 생성

가능성이 활발히 연구되고 있으며 프로그래밍 교육 측면에서 코드 생성, 디버깅과 같은 교수·학습 환경에 직접적으로 도움을 줄 수 있는 관점에서의 연구들이 유의미한 결과를 도출하고 있다[14-17]. 이에 본 연구는 기존의 연구 결과를 발전시켜 LLM을 활용하여 프로그래밍 교육 콘텐츠를 생성하고 기초 학습자의 프로그래밍 교육 시 발생할 수 있는 어려움을 해결하는데 도움을 줄 수 있는 환경을 구축하며 적용 가능성을 확인하고자 한다.

2. 관련 연구

2.1 프로그래밍 교육 방법

컴퓨팅 사고력을 향상 시킬 수 있는 효과적인 방법으로 프로그래밍이 강조됨에 따라 학습자나 교육자 겪는 프로그래밍 교육의 어려운 점에 관한 연구가 지속적으로 이루어졌다. 먼저 학습자에게 비실용적인 교재와 강의 형태의 수업, 다수의 학생을 대상으로 수업이 이루어지는 학급 형태, 개별 학생에게 발생하는 불평등과 같은 교수·학습 환경에서 발생하는 어려운 점이 강조 되었다[7,17,18]. 또한 프로그래밍 활동 시 발생하는 오류나 코드 자체의 추상성으로 인한 인지적 부담, 코드를 읽고 결과를 예상하는 능력의 부족, 프로그래밍에 대한 흥미 부족 등 학습자 내적 문제로 인한 어려움 등이 선행 연구를 통해 제시되고 이를 극복하기 위한 다양한 교육 방법이 개발되었다[8,10,20].

이후 다양한 연구자를 통해 프로그래밍 교육의 효과성을 높이기 위한 교육의 방향성 관련 연구가 이루어졌으며 주요 선행 연구에 따르면 프로그래밍 학습을 통해 기대하는 바를 명확히 하고 중요성과 가치를 인식시키는 과정이 필요하며 학습자 중심의 교수·학습 모델을 통해 과제에 대한 적극적인 참여와 긍정적인 성공의 기대를 높일 수 있는 평가 방향이 중요함을 확인할 수 있다[11]. 또한 교육의 주요 소재로 학생들의 삶의 맥락과 관련 있는 주제의 선정이 필요하며 흥미로운 사례 포함을 통해 참여를 촉진하고 능동적인 의사소통과 실습 중심의 환경 구성이 필수적으로 강조되었다[7,21]. 그리고 학생들에게 제공되는 교육자료는 간결하고 추상적인 코드보다는 가독성이 높고 교육을 목적으로 설계된 형태로 제시되어야 하며 구문적 혹은 개념적 지식에 대한 학습이 선행되어야 함을 확인할 수 있다[3,9].

프로그래밍 교육의 순서와 실습 방법을 중심으로 진행된 선행 연구에서는 코드 추적(코드의 실행 결과 예측), 코드 조작(재구성 및 재배열), 코드 작성의 순서로 프로그래밍 능력이 발달하며 해당 순서로 교육이 이루어져야 함을 확인할 수 있다. 또한 교수·학습 환경에서 이를 적용하기 위해 프로그래밍 템플릿을 제공하거나 단계적으로 일부가 가려진 소스 코드를 활용하는 방법 및 주어진 코드 조각을 문제 해결 알고리즘에 맞게 재배열하는 파슨스 문제(Parsons Problems) 구성 방법 등이 개발되었고 프로그래밍 기초 학습자에게 적용하여 그 효과성을 확인한 연구들이 이루어졌다[22-25].

이러한 교육 방법의 현장 적용을 위해서는 다양한 프로그래밍 예제 및 문제 개발이 필수적으로 선행되어야 한다. 하지만 교수자 개인이 모든 리소스를 개발하는데 한계가 있으며 학생 등 외부에서 생성된 리소스 또한 품질과 관련된 문제가 발생할 수 있기 때문에 양질의 리소스 개발을 위한 방법적인 측면에서의 논의가 필요함을 도출할 수 있다[12,26].

2.2 대형 언어 모델(LLM)과 프로그래밍 교육

LLM은 일반적으로 대량의 텍스트를 중심으로 사전 학습된 모델을 프롬프트(Prompt)라는 입력을 통해 활용할 수 있도록 구성되어 있다. 이때 프롬프트는 모델에게 특정 지시를 내리기 위한 자연어이며 모델의 내부 구조나 추론 과정이 사용자에게 명시적으로 안내되지 않기 때문에 프롬프트 엔지니어링(Prompt Engineering)이라 불리는 다양한 접근 방식을 적용할 수 있다[27,28].

최근 코드 생성에 특화된 모델뿐만 아니라 범용적으로 활용 가능한 LLM을 활용하여 코드를 생성하고 코드의 양적 질적 평가를 통해 활용 가능성을 다각도 분석하여 프로그래밍을 실질적으로 지원할 수 있는 방법이 활발히 논의되었다[27,28]. 특히 프로그래밍을 위한 코드 생성뿐만 아니라 교육을 위한 목적에서 교수·학습 환경에서 활용 가능한 예제를 생성하는 방안이 연구되었다. 주요 연구로 LLM을 활용하여 코드를 설명하는 문장을 생성하고 이를 학생이 생성한 문장과 비교하여 교육 현장에 적용 가능함을 확인한 연구 결과를 확인할 수 있다. 또한 다양한 유형의 프로그래밍 문제를 해결할 수 있는 코드 생성 기능을 통해 LLM이 직접 문제 및 해결 가능한 코드를 생성하여 교육적으로 활용 가능하다는 결론을 도출한 연구가 뒤

이어 이루어졌다[12,14,29] 그리고 프로그래밍 교육에서 즉각적인 피드백 제공으로 교수자를 지원하며 학생들에게 힌트를 제공하기 위해 LLM을 활용한 자동 힌트 생성 시스템을 구축하고 교육 현장에 적용 가능성을 확인한 연구도 유의미한 결과를 도출하였다[15,13]

LLM을 프로그래밍 교육에 적용한 선행 연구들은 개별 학습자를 지원하거나 콘텐츠를 생성하는 측면에서 교육의 보조 도구로서의 높은 활용성에 대한 긍정적인 측면과 함께 일부 한계점을 강조하고 있다. 주요 한계점을 살펴보면 프로그래밍 과정 속에서 학생이 LLM과 상호 작용하는 주요 패턴 분석을 통해 LLM이 생성한 콘텐츠에 의존하며 잘못 생성된 결과물로 올바른 솔루션에서 계속 멀어지는 상황을 겪는 문제점을 확인하였다. 이를 통해 LLM 활용 시에 과도한 의존은 프로그래밍 교육에 악영향을 미칠 수 있음을 주장하였다[13]. 또한 여러 선행 연구를 통해 LLM은 학습 상황에서 학생들에게 직접적인 답변이나 학습자가 직접 작성해야 하는 코드 등을 제시하여 학습 효과를 저해하는 문제점을 보임을 확인할 수 있다[15,30]. LLM의 응답을 질적으로 분석한 연구에서는 프롬프트의 형태에 따른 응답의 질적 차이가 크게 나타나는 문제점을 제시하였으며 생성된 코드의 길이가 학생에게 적절치 않아 가독성이 떨어지고 코드의 추상성이 높아 일부 생성 콘텐츠는 기초 학습자에게 적절하지 않다는 결론을 도출하였다[12,14]. 또한 프로그래밍 교육에 LLM을 적용한 다수의 연구들은 LLM 모델의 API를 활용하여 교육 콘텐츠를 제공하는 방법을 활용하였다. 이에 서비스를 운영하기 위한 유지 보수 비용과 API 활용으로 사용량에 따라 함께 증가하는 비용 문제가 발생하여 대규모 교육 환경 및 일반적인 교육 상황에서 지속적으로 활용할 수 없다는 한계를 보였다. 또한 구조화된 인터페이스로 인해 모든 환경에 적용할 수 없다는 유연성 부족 문제가 함께 제시되기도 하였다[15,16,30].

선행 연구를 종합하면 LLM은 프로그래밍 교육 교수·학습 환경에서 교수자를 보충 지원할 수 있는 도구로서 높은 활용성을 가지고 있지만 교육의 목적에 따라 체계적으로 설계된 프롬프트나 구체적인 목적을 가진 접근이 필요함을 도출할 수 있다. 이에 본 연구는 사용자의 과도한 의존으로 인한 한정적인 상호작용을 해결하기 위해 학습자가 주도적으로 학습하고 문제를 해결할 수 있는 콘텐츠 생성에 목적을 두며 교육학적 관점에서 효과적인 프로그래밍 교육 방법을

적용할 수 있도록 한다. 또한 연구 결과물의 현장 파급력을 높이기 위해 대규모 교육 현장에 적용 가능하며 장기간 운영 가능한 플랫폼을 활용하여 지속 가능한 프로그래밍 교육 지원 체계를 구성하고자 한다.

3. 연구 방법

3.1 프로그래밍 교육용 GPTs 챗봇 설계

프로그래밍 교육 콘텐츠 생성을 위한 LLM은 코드 생성 능력이 다른 모델 대비 비교적 뛰어나며 접근성이 높은 OpenAI 사의 GPT-4o 모델과 함께 GPTs 플랫폼을 활용한다[31,32]. GPT-4o 이전의 모델인 GPT-4 및 GPT-3.5 모델에서도 정확도 높은 코드 생성 능력을 보여 다양한 선행 연구에서 활용되었으며 최근 업데이트로 개인화된 LLM 서비스를 구성하여 제공할 수 있는 GPTs 플랫폼이 무료 사용자도 활용할 수 있도록 확대되어 학생들의 접근성을 보다 높일 수 있다는 장점이 있다. 또한 API를 활용한 서비스와 달리 GPTs를 활용하는 경우 서비스 유지 보수 비용이 추가적으로 발생되지 않기 때문에 지속적으로 운영할 수 있으며 간단한 URL을 통해 학생들에게 쉽게 제공할 수 있다는 장점이 있다[33].

본 연구를 통해 구성될 GPTs는 교수자를 보충 지원하는 챗봇으로 교수·학습 전체 과정에서 개별 학습자의 프로그래밍 개념학습 및 실습을 지원하는 보조 자료 역할과 자유 실습 시 효과적인 교육 콘텐츠를 제공하는 역할을 하도록 설계한다. 또한 프로그래밍 교육 선행 연구를 바탕으로 세부적인 기능은 크게 프로그래밍 개념학습, 프로그래밍 연습, 프로그래밍 도움으로 나누어 구성하였다. 일반적인 대화 형식으로 이루어지는 GPTs의 UI 특성을 고려하여 문제 해결 프로그래밍과 프로그래밍 도움은 개별 IDE를 통해 이루어질 수 있도록 하였다. 전체적인 GPTs의 구성은 Figure 1과 같다.

GPTs에서 활용되는 프로그래밍 언어는 교육의 목적에서 주로 활용되는 Python(파이썬)과 C언어로 설정하였다. 프로그래밍의 주요 개념은 2022 개정 교육과정 중등 및 고등 정보 과목과 과학 계열 선택 과목인 정보 과학에서 명시적으로 제시하고 있는 주요 개념을 도출하였으며 파이썬과 C언어 관련 기술 서적에서 제시하고 있는 개념을 중심으로 Table 1과 같이 정리하고 공통적으로 제시하는 개념을 활용하였다.

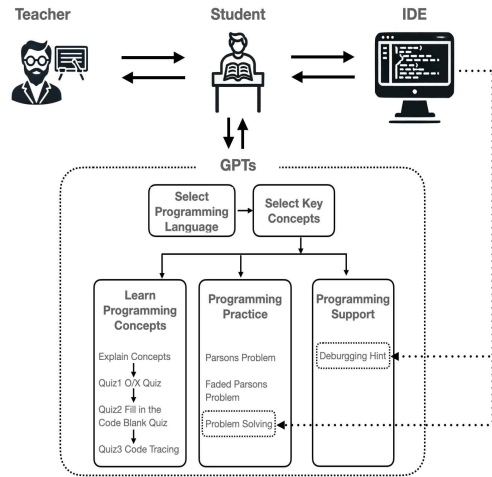


Figure 1. Designing GPTs for Programming Education

Table 1. Programming Key Concepts

Category		Key Concept
2022 Revised Curriculum	Middle School Informatics	Input, Processing, Array, List, Logical Operation, Nested Control Structures, Function
	High School Informatics	Data Type, Standard Input/Output, File Input/Output, Multidimensional Data Structures, Various Control Structures, Class, Instance
	High School Information Science	Function (Recursion), Array, List, Struct, Pointer
Python Programming Book	A	Input/Output, Data Types (Numeric, Character, String, List, Tuple, Dictionary, Set, Boolean), Variable, if, while, for, Function, Class
	B	Output, Data Types (Numeric, Character, String), Operator, Variable, Input/Output, List, Dictionary, Tuple, if, while, for, Function, Class
C Programming Book	C	Variables and Constants, Data Types, Input/Output, Loop, Conditional Statement, Function, Array, String, Pointer, Struct
	D	Input/Output, Data Types, Constants and Variables, Function, Operator, Conditional Statement, Loop, Array, Pointer, Struct, Union, Enum, Dynamic Memory

GPTs 플랫폼은 특정 기능에 대한 자연어 설명을 통해 지침을 구성할 수 있도록 기능을 제공하고 있다 [34]. 하지만 입력되는 프롬프트가 답변의 질에 크게 영향을 미치며 세션에 따른 응답이 변하게 되는 LLM의 특성을 고려하여 Figure 2와 같이 지침 개발을 위한 과정을 설계하였다.

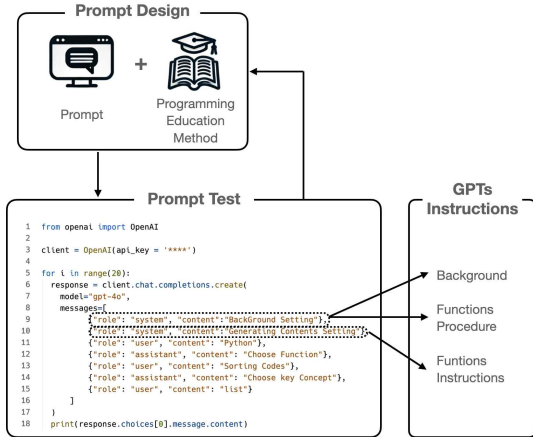


Figure 2 Configuring Instructions with Prompt Testing

먼저 프로그래밍 교육 방법을 적용한 프롬프트를 설계하고 API를 활용하여 각 기능별 프롬프트를 입력한 뒤 응답을 반복적으로 생성하여 결과를 분석하였다. 이를 통해 GPTs의 지침을 세부적으로 조정할 수 있도록 하였다. 면밀한 프롬프트 테스트를 위해 GPTs와 동일한 GPT-4o 모델을 활용하였으며 지침 구성에 전체적인 조건에 해당하는 배경과 프로그래밍 학습 콘텐츠 생성 조건을 각각 API의 system 항목으로 설정하였다. 또한 GPTs와의 의사소통 단계에 맞추어 사용자의 요청을 user에 할당하고 모델의 응답을 system에서 설정한 것과 동일한 응답으로 assistant에 할당하여 대화 시나리오를 구성하였다. 그 외 답변의 형태를 결정하는 temperature 등의 파라미터는 기본적으로 설정하고 20회 반복하여 생성한 뒤 결과를 확인하였다[34]. 최종적으로 가장 적절한 결과를 생성하는 프롬프트를 선택하고 반복적으로 수정하여 GPTs의 지침을 구성하였다.

3.2 프로그래밍 교육용 GPTs 챗봇 검증 방법

최종적으로 구성된 GPTs의 교육적 타당성을 확인하기 위해 11명으로 구성된 컴퓨터 교육 관련 전문가 집단을 구성하여 각각의 세부적인 기능에 대한 교육적 적합도, 챗봇 응답의 정확도 및 주관식 의견을 수렴하여 반영하도록 하였다. 연구에 참여한 컴퓨터 교육 전문가의 상세 정보는 Table 2와 같다.

Table 2. Information of Experts Group

	Category	Experts (%)	Total
Education Targets	Elementary school	4 (36.3)	11
	Meddle school	1 (9.1)	
	High school	3 (27.3)	
	University	3 (27.3)	
Education Experience	5~10 years	1(9.1)	11
	10~15 years	2 (18.3)	
	More than 15 years	8 (72.6)	
Final Major	Computer Education	10 (90.9)	11
	Computer Science	1 (9.1)	
Final Degree	Master	5 (45.5)	11
	Doctor	6 (54.5)	

설문은 각 기능의 구성에 대한 응답의 강도보다는 전문가의 찬반 의견이 중요하므로 중립적인 응답을 배제하고 ‘매우 동의함’ 및 ‘매우 정확함’ 4점에서 ‘전혀 동의하지 않음’ 및 ‘전혀 정확하지 않음’ 1점의 4점 척도로 응답할 수 있도록 설계하고 온라인으로 실시하였다. 그리고 전문가의 의견이 긍정적으로 수렴되었는지 여부를 양적으로 판단하기 위해 CVR(Content Validity Ratio) 값을 도출하며 11명의 전문가 수에 따른 최소 기준인 0.59를 기준으로 평가하였다. 상세한 분석을 위해 추가적으로 전문가 50%가 응답한 구간을 수치화한 지표인 합의도(Consensus)를 도출하여 0.75 이상을 기준으로 해석하며 전문가의 의견 일치 정도를 양적으로 확인하기 위한 수렴도(Convergence)를 도출하여 0.50 이하를 기준으로 확인하였다. 각각의 지표에 대한 공식은 아래와 같다[35-38].

$$CVR = \frac{N_e - \frac{N}{2}}{\frac{N}{2}}, \quad Consensus = 1 - \frac{Q_3 - Q_1}{Mdn}$$

$$Convergence = \frac{Q_3 - Q_1}{2}$$

N_e : 긍정응답 배열 수, Q_1 : 1분위 수, Q_3 : 3분위 수, Mdn : 중앙값

4. 연구 결과

4.1. 프로그래밍 교육용 GPTs 챗봇 구성 결과

GPTs에 각 기능과 전체적인 생성 조건에 따른 지침 구성 요약은 Table 3과 같다.

Table 3. Configuring GPTs Instructions

Step	Instructions
Background	Role: ▷ Chatbot to help teach C and Python programming Goal: ▷ Help users understand and practice programming concepts and get help when needed Response style : ▷ Friendly and encouraging tone ▷ Minimise technical terms and provide examples based on real-life subjects ▷ Does not provide the correct answer to the problem posed, even if the learner requests it
	Choosing a programming language: Prompted to choose between C and Python Select key programming concepts: ▷ Prompted to choose 1 concept C language concept: ▷ Output, types, constants and variables, operators, conditional statements, looping, functions, local and global variables, arrays, strings, pointers, memory allocation, structs, unions, enumerations Python concepts: ▷ Output, data types, operators, variables, input/output, lists, dictionaries, tuples, if, while, for, functions, classes and modules Choose a function: ▷ Learn programming concepts, Programming Practice, Programming Support.
Function Procedure	Learn Programming Concepts
	▷ Explain the need for and benefits of the concept using material close to home ▷ Provide example code with comments to illustrate key concept ▷ Provide O/X quizzes, code fill-in-the-blank, and code tracing questions, in that order, to assess understanding of the concept
	Programming Practice
	Choose 1 of the following types of problems: ▷ Arrange code, Arrange code with blanks, Problem Solving ▷ Provide hints if the user enters an incorrect answer to a provided problem, encouraging them to find the correct answer on their own ▷ Arrange Codes: Parsons Problems - List the provided code snippets in the correct order ▷ Arrange Codes with blank: Faded Parsons problems - List the code snippets in the correct order with some blanks ▷ Problem solving: Presenting a problem in storytelling format, centred around something from your life, and writing code to solve it
	Programming Support
	Provide hints to help users troubleshoot the code they enter. ▷ Giving advice and steps without presenting the solution to the problem as modified code.
Function Instructions	

선행 연구 분석을 통해 LLM을 활용한 교육 콘텐츠의 생성에 있어서 사용자 요청 시 생성된 문제 혹은 퀴즈에 대한 해답이 직접적으로 제시될 수 있다는 문제를 확인하였다. 이에 GPTs의 응답 스타일에 직접적인 정답을 제시하지 않는다는 지침을 포함하였다 [13,15,29]. 또한 프로그래밍 기초 학습자를 고려하여 전문 용어의 사용을 최소화하며 생활 속 소재를 중심으로 한 예시를 활용할 수 있도록 구성하였다[21].

프로그래밍 학습의 주요 기능 중 개념 학습의 경우 학습자가 선택한 프로그래밍 언어와 개념에 대한 기본적인 원리, 해당 개념을 사용하는 이유, 기본 구조, 예시 코드를 생활 속 주제를 중심으로 설명하도록 구성하였다. 개념 학습 콘텐츠 생성 예시는 Figure 3과 같다.

What is a `for` loop?

A `for` loop is a control flow statement for specifying iteration, which allows code to be executed repeatedly. It is commonly used to iterate over the elements of a sequence (like a list, string, or tuple).

Why use a `for` loop?

- It allows you to perform repetitive tasks efficiently.
- It simplifies handling elements in lists, strings, and other iterable objects.
- It improves code readability.

Basic Structure

The basic structure of a `for` loop in Python is as follows:

```
python
for variable in sequence:
    code to execute
```

Here, `variable` is a temporary variable that takes the value of each element in the sequence, and `sequence` is the iterable object (like a list, string, tuple, etc.).

Example Code

```
python
# Example of printing each element in a list
fruits = ['apple', 'banana', 'cherry']
for fruit in fruits:
    print(fruit) # Prints the name of each fruit
```

This code assigns each element of the list `fruits` to the variable `fruit` one by one and prints it. The output will be:

```
python
apple
banana
cherry
```

Here's another example where each character of a string is printed:

```
python
# Example of printing each character in a string
for char in "hello":
    print(char) # Prints each character
```

Figure 3. Example of Generating Learning Programming Concept Contents

프로그래밍 개념이 제시된 후에는 Figure 4와 같이 학습자의 이해도를 확인할 수 있는 O/X 퀴즈, 빈칸 채우기 퀴즈를 제시하며 프로그래밍 기술 중 가장 먼저 발달하는 것으로 알려진 코드 추적 기술을 고려하여 코드 실행 결과를 확인하는 퀴즈를 각각 순차적으로 제시하도록 하였다[17].

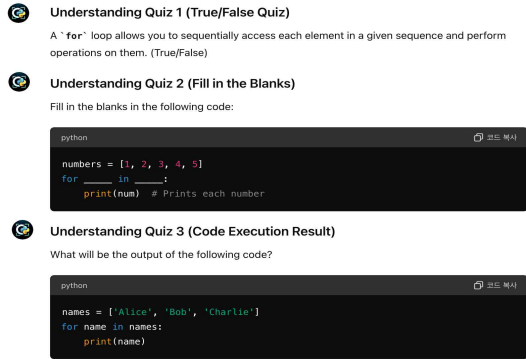


Figure 4. Quiz in Learning Programming Concept

프로그래밍 연습은 기초 학습자를 위해 프로그래밍 기능 발달 순서를 고려하여 다양한 유형의 콘텐츠를 단계적으로 제공할 수 있도록 하였다[25]. 코드 나열하기는 선행 연구 분석을 통해 학습자들이 겪는 주요 문법 오류의 가능성을 줄여 줄 수 있으며 논리적 사고를 촉진하고 문법의 이해를 증진시켜 줄 수 있는 파스너스 문제 형태로 제시하였다. 단순 기계적 반복을 통해 답을 도출하는 파스너스 문제 유형의 한계점을 최소화하고자 잘못된 코드 라인을 추가하여 선택적으로 나열할 수 있도록 하였다. 또한 문제에 대한 응답은 채팅창을 통해 코드 조각의 번호를 올바른 순서로 입력하는 형태로 답할 수 있도록 구성하였다. 코드 나열하기 문제의 예시는 Figure 5와 같다. 빈칸이 포함된 코드 나열하기는 심화된 형태의 파스너스 문제 유형으로 빈칸을 포함하여 제시하도록 구성하였으며 단계적으로 난이도를 높여 실습하도록 하였다.

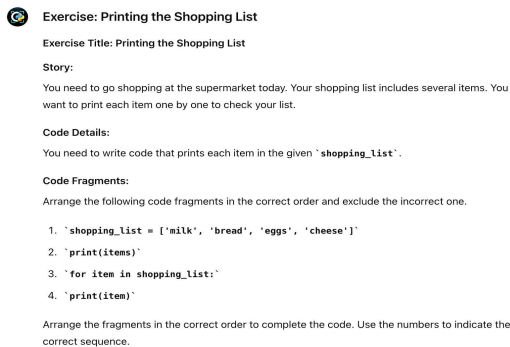


Figure 5. Example of Arrange Code Programming Practice

문제 해결은 실생활의 문제 상황을 제시하고 이를 해결할 수 있는 코드를 입력하는 형태로 구성하였으

며 챗봇의 특성상 코드 편집 기능은 제공되지 않으므로 전체 코드 작성은 외부 IDE를 통해 작성하고 결과만 입력할 수 있도록 안내하였다. 또한 문제 제시와 함께 코드 작성에 도움을 줄 수 있도록 주요 개념과 입력 및 출력에 대한 안내가 함께 제시되도록 구성하였다. 문제 해결 유형의 생성 예시는 Figure 6과 같다.

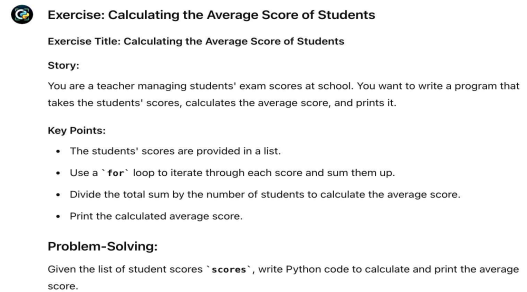


Figure 6. Example of Problem Solving Programming Practice

프로그래밍 도움은 학습자가 작성한 코드를 입력하고 오류나 코드 개선 방향 등을 요청할 수 있는 기능이 다. 외부 IDE를 통해 작성된 코드를 입력할 수 있도록 안내하며 직접적인 코드의 수정 결과를 제공하지 않고 해결에 필요한 조언과 단계를 제시하도록 구성하였다. 사용자의 코드 입력에 대한 오류 수정 도움의 생성 결과는 Figure 7과 같다.

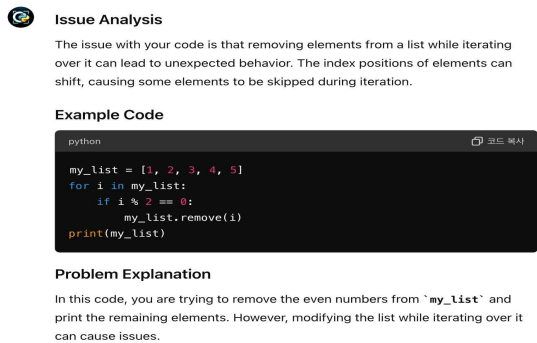


Figure 7. Example of Programming Support

4.2. 전문가 그룹 1차 적합성 평가 결과

구성된 챗봇 기능의 교육적 적합성과 응답 결과의 정확성에 대한 전문가 검토 결과를 활용하여 평균과 CVR, 합의도, 수렴도를 Table 4와 같이 도출하였다.

Table 4. Experts Group Validity Assessment Results(1st)

Evaluation Items	Mean	CVR	Consensus	Convergence
Programming Language (C/Python)	4.00	1.00	1.00	0.00
C Language Key Concepts	3.18	0.64	0.67	0.50
Python Key Concepts	3.18	0.64	0.67	0.50
User Experience Flow	3.82	1.00	1.00	0.00
Easy Examples and Real-World Material	3.64	1.00	0.75	0.50
No Right Answers	3.73	1.00	0.88	0.25
Generating Concepts Contents	3.64	1.00	0.75	0.50
Accuracy of Responses	3.27	1.00	0.83	0.25
O/X Quiz	3.27	1.00	0.83	0.25
Accuracy of Responses	3.27	0.64	0.67	0.50
Fill in the Code Blank Quiz	3.73	1.00	0.88	0.25
Accuracy of Responses	4.00	1.00	1.00	0.00
Code tracing	3.64	1.00	0.75	0.50
Accuracy of Responses	4.00	1.00	1.00	0.00
Arrange Code	3.82	1.00	1.00	0.00
Accuracy of Responses	2.64	0.09	0.67	0.50
Arrange Code with Blank	3.73	1.00	0.88	0.25
Accuracy of Responses	2.91	0.45	0.83	0.25
Problem Solving	3.18	0.45	0.50	0.75
Accuracy of Responses	3.82	1.00	1.00	0.00
Programming Support	3.82	1.00	1.00	0.00
Accuracy of Responses	3.91	1.00	1.00	0.00

챗봇의 전체적인 배경과 관련되는 프로그래밍 언어, 언어별 개념, 사용자 경험 흐름, 쉬운 예제 사용, 정답을 제공하지 않음은 모두 적합한 것으로 나타났다(CVR>0.59). 상대적으로 합의도가 다소 낮게 나타난 각 언어 별 주요 개념의 적합도(Consensus=0.67)에 대한 의견을 살펴보면 ‘일부 개념은 기초 학습자에게 어려운 내용으로 제외가 필요함’, ‘제시된 개념 이외에 추가적으로 학습을 원하는 개념이 있는 경우가 있기 때문에 개념을 제시할 때 예시 형태로 제시할 필요가 있음’ 등을 확인할 수 있었다.

프로그래밍 개념 학습과 관련된 응답을 살펴보면 모든 세부 기능에 대해 기능의 적합성과 응답의 정확성이 기준을 만족하는 것으로 나타났다(CVR>0.59). 비교적 낮은 합의도를 보였던 O/X 퀴즈 응답의 정확성(Consensus=0.67)에 대한 의견을 살펴보면 ‘사용자가 오답을 입력하는 경우 바로 정답을 제시하는 경우가 자주 나타남’, ‘O/X 퀴즈 문제는 오답이 입력되는 경우 오답에 대한 설명 이후 다음 단계로 넘어가야 함’ 등을 확인할 수 있었다.

프로그래밍 연습의 경우 코드 나열하기와 빈칸이 포함된 코드 나열하기의 세부 기능은 교육적으로 적합한 것으로 나타났으나 문제 해결하기는 적합하지

않은 것으로 나타났다(CVR<0.59). 관련된 주요 의견으로 ‘기초 학습자가 코드를 직접 작성하기 어려움’, ‘기초 학습자가 프로그래밍으로 문제를 해결하기에는 제공되는 정보가 너무 부족함’ 등이 나타났다. 각 세부 기능 응답 정확도의 경우 코드 나열하기와 빈칸이 포함된 코드 나열하기가 낮은 CVR 값으로 보여 적합하지 않은 것으로 나타났다(CVR<0.59). 주요 의견을 살펴보면 ‘각각의 코드 라인을 표현하는 방법이 세션마다 다르게 나타나 혼란스러움’, ‘빈칸에 들어갈 코드와 코드 라인의 순서를 기입하는 방법이 제시되지 않음’, ‘문제 해결을 위한 힌트를 요청하는 경우 반복된 요청에도 전체적으로 비슷한 힌트만 계속 제시함’ 등을 확인할 수 있었다.

프로그래밍 도움의 경우 기능의 적합성과 응답의 정확성 모두 기준을 만족하는 것으로 나타났으며 개선을 위한 구체적인 의견은 제시되지 않았다. 마지막으로 챗봇의 전체적인 구성에 대한 추가적인 의견으로 ‘기본적인 학습을 선택하기 위한 상호작용 단계가 너무 세부적으로 나누어져 있음’, ‘언어 모델 사용 한계가 빨리 나타나서 교수·학습 환경에서 오랜 시간 사용할 수 없음’, ‘같은 질문에도 너무 다양한 포맷의 코드나 텍스트가 생성되어 통일이 필요함’ 등을 확인할 수 있었다.

4.3. 프로그래밍 교육용 GPTs 챗봇 재구성

전문가 검토 결과를 통해 기존 챗봇의 개선점을 도출하고 사용자 경험 흐름과 세부 기능에 대한 지침을 변경하였다.

먼저 챗봇 이용 시 발생하는 GPT-4o 모델의 사용량을 줄이기 위해 상호작용 횟수를 줄이도록 의사소통 과정을 변경하였다. GPTs 사용 방법 제시를 통해 세부 기능과 기능 선택에 대한 내용을 한꺼번에 안내할 수 있도록 하였으며 사용자는 프로그래밍 언어, 주요 개념, 학습 기능을 한꺼번에 입력하고 학습을 진행할 수 있도록 구성하였다. 또한 GPTs를 구성하고 있는 전체적인 지침 중 반복적이거나 간략하게 표현 가능한 지침을 수정하여 토큰 수를 기존의 2,668에서 2,423로 줄여 더 많은 활용이 가능하도록 구성하였다. 챗봇 실행을 위한 사용 방법 안내 결과는 Figure 8과 같다.

- Provides various features to support learning C language and Python programming.
- **Examples of Key Concepts in C Language:** Basic Output, Data Types, Constants and Variables, Operators, Conditional Statements, Loops, Functions, Local and Global Variables, Arrays, Strings, Pointers, Memory Allocation, Structures, Unions, Enumerations
- **Examples of Key Concepts in Python:** Output, Data Types, Operators, Variables, Input/Output, Lists, Dictionaries, Tuples, Conditional Statements - if, Loops - while, Loops - for, Functions, Classes and Modules
- **Main Features:**
 - **Learning Programming Concepts:** Provides quizzes to check understanding of the input programming key concepts.
 - **Programming Practice - Arrange the Code:** Arrange the given code in the correct order.
 - **Programming Practice - Fill in the Blanks and Arrange the Code:** Fill in the blanks in the given code and arrange the code in the correct order.
 - **Programming Practice - Problem Solving:** Write code to solve real-life problems.
 - **Programming Help:** Get help solving problems encountered while writing code.
- For learning programming, please enter **Programming Language - Concept - Main Feature** in order. Example: Python - List - Arrange the Code

Figure 8. Manual for Reduce the Number of GPTs Interactions

세부 기능 실행을 통해 생성된 콘텐츠의 제시 방법에 대한 통일성을 높이기 위해 답변 및 콘텐츠 생성을 위한 상세한 지침과 더불어 ‘예시 템플릿’을 마크다운(Markdown) 양식을 사용하여 구성하였으며 그 예시는 Table 5와 같다.

Table 5. Example Response Template

```

### Programming Practice - Arrange code:
- Provide the following exercises in which students are asked to list the numbered lines of code in the correct order in a Markdown code form centred on a selected concept.
- Example template :
- Code Details : [Provide a detailed flow of the code to be presented in the problem before presenting the code lines].
- Code : [When listing the code snippets from the problem, follow the principles of Parsons Problems, including incorrect code]
- Put the numbers of the correct codes in order, excluding the incorrect codes.
    
```

프로그래밍 개념 학습의 경우 개념 안내와 이해도 확인을 위한 퀴즈 유형 3가지가 동시에 제시되도록 구성하였으며 사용자는 순차적으로 정답을 입력하여 학습할 수 있도록 구성하여 상호작용 횟수를 줄였다.

프로그래밍 연습의 코드 나열하기와 빈칸을 포함한 코드 나열하기 문제의 경우 챗봇의 응답 중 코드를 제시하는 방법이 세션마다 달라 사용자에게 혼동을 줄 수 있다는 전문가의 의견을 반영하여 각각의 코드 라인을 마크다운의 코드 양식 형태로 제시되도록 지침에 명시하였다. 또한 코드 나열을 위한 추가 정보를 제공하기 위해 제시된 코드의 올바른 실행 흐름에 대한 텍스트 설명을 추가하여 구성하였다. 개선된 코드 나열하기 유형의 프로그래밍 연습 예시는 Figure 9와 같다.

Python List - Arrange the Code

Code Details:
The following code snippet demonstrates how to store student test scores in a list, add and modify scores, calculate the average score, and print the result. The code's flow is as follows: creating the list, adding scores, calculating the average score, and printing the result.

Code:

```

python
1. average_score = sum(scores) / len(scores)
2. scores.append(80)
3. scores = [50, 90, 70, 90]
4. print("The average score of the students is:", average_score)
5. scores.remove(70)
6. scores.insert(1, 80)
    
```

Exclude the incorrect code and enter the correct code numbers in order.

Figure 9. Example of Improved Arrange Code Contents

문제 해결 유형의 경우 이야기와 프로그래밍의 주요 방향, 문제를 해결하는데 도움이 되는 알고리즘을 제시하는 도움말로 나누어 생성할 수 있도록 구성하였다. 또한 사용자가 힌트를 요청하는 경우 같은 응답만 반복하는 문제점을 개선하기 위해 선행 연구 분석을 통해 프로그래밍을 도울 수 있는 힌트의 유형으로 추상적이며 사용자가 집중해야 할 부분을 지시하는 ‘방향 힌트’와 직접적으로 작성해야 할 코드의 개념을 제시하는 ‘도구적 힌트’로 나누어 순차적으로 제시할 수 있도록 구성하였다[16]. 개선된 문제 해결 생성 예시는 Figure 10과 같다.

Programming Practice - Problem Solving: Python - Lists

Story:

You and your friends are planning a movie marathon. Each friend has created a list of movies they want to watch. Your task is to combine these lists into one master list and ensure there are no duplicate movies.

Problem:

Write a program that satisfies the following requirements:

1. Create a list of movies for each friend.
2. Combine all friends' lists into one master list.
3. Remove duplicate movies from the master list.
4. Print the final movie list.

Hints:

1. To combine multiple lists, you can use the '+' operator or the 'extend()' method.
2. To remove duplicates, you can convert the list to a set and then back to a list.

Think about the algorithm needed to solve this problem and write the code. If you need help, feel free to ask for hints!

Figure 10. Example of Improved Problem Solving Contents

프로그래밍 도움은 입력된 코드에 대해서 ‘방향적 힌트’와 ‘도구적 힌트’를 순차적으로 생성하며 직접적으로 수정된 코드를 제시하지 않도록 구성하였다. 개선된 프로그래밍 도움의 생성 예시는 Figure 11과 같다.

Hint Type: Directional Hint

When you iterate through a list and remove elements from it, the indices can get messed up, causing some elements to be skipped. Consider using a copy of the list for iteration or iterating through the list in reverse order when removing elements.

If this hint is not enough, feel free to ask for a more detailed hint!

Hint Type: Instrumental Hint

Instead of modifying the original list while iterating through it, consider using a filtering approach to create a new list that only contains the elements you want to keep. For example, you can use the 'filter()' function or a list comprehension.

Here's an example using the 'filter()' function:

```
python
my_list = [1, 2, 3, 4, 5]
my_list = list(filter(lambda x: x % 2 != 0, my_list))
print(my_list)
```

This approach will safely remove the even numbers from the list.

If you need further assistance, please let me know!

Figure 11. Example of Improved Programming Support Contents

4.4. 전문가 그룹 2차 적합성 검토

수정된 GPTs 챗봇을 활용하여 2차 전문가 검토를 진행하였으며 2차 전문가 검토 응답을 정리한 결과는 Table 6과 같다.

Table 6. Experts Group Validity Assessment Results(2nd)

Evaluation items	Mean	CVR	Consensus	Convergence
Programming Language (C/Python)	4.00	1.00	1.00	0.00
C Language Key Concepts	3.73	1.00	0.88	0.25
Python Key Concepts	3.73	1.00	0.88	0.25
User Experience Flow	3.36	1.00	0.67	0.50
Easy Examples and Real-World Material	4.00	1.00	1.00	0.00
No Right Answers	3.73	1.00	0.88	0.25
Generating Concepts Contents	4.00	1.00	1.00	0.00
Accuracy of Responses	3.73	1.00	0.88	0.25
O/X Quiz	3.82	1.00	1.00	0.00
Accuracy of Responses	3.73	1.00	0.88	0.25
Fill in the Code Blank Quiz	3.91	1.00	1.00	0.00
Accuracy of Responses	3.64	1.00	0.75	0.50
Code tracing	3.82	1.00	1.00	0.00
Accuracy of Responses	3.82	1.00	1.00	0.00
Arrange Code	3.82	1.00	1.00	0.00
Accuracy of Responses	3.82	1.00	1.00	0.00
Arrange Code with Blank	3.64	1.00	0.75	0.50
Accuracy of Responses	3.64	1.00	0.75	0.50
Problem Solving	3.64	1.00	0.75	0.50
Accuracy of Responses	3.82	1.00	1.00	0.00
Programming Support	4.00	1.00	1.00	0.00
Accuracy of Responses	4.00	1.00	1.00	0.00

모든 항목에 있어서 교육적 적합성과 응답의 정확도가 기준치 이상으로 나타났다(CVR>0.59). 상대적으로 사용자 경험 흐름에 대한 합의도가 낮게 나타났는데 상세 이유를 살펴보면 교육 콘텐츠 생성을 위해 순차적으로 메뉴 입력이 이루어지던 과정에서 한꺼번에 입력하는 형태로 수정되었고 이로 인한 사용의 불편함이 일부 전문가를 통해 제시되었다. 이에 학생 및 교사의 사용에 대한 이해를 돕기 위해 GPTs 실행 시 제시되는 기본 질문에 ‘사용 방법’, ‘프로그래밍 개념 학습’, ‘프로그래밍 연습’, ‘프로그래밍 도움’ 버튼을 생성하고 해당 의견을 제시한 전문가 검토를 통해 적합성을 확인하고 설문을 종료하였다.

최종적으로 완성된 GPTs의 구성 지침은 부록에서 확인할 수 있다. 또한 챗봇의 이름을 ‘Programming Study Tutor’로 설정하고 GPTs 플랫폼에 전체 공개로 배포하였다.

5. 결론 및 제언

본 연구는 컴퓨팅 사고력 증진 및 AI 교육을 위한 프로그래밍 교육의 중요성을 확인하고 선행 연구 분석을 통해 프로그래밍 교육 기초 학습자를 위한 교육 콘텐츠 생성 GPTs 챗봇을 구성하였다. 연구의 결과물은 다음과 같은 교육적 의의를 갖는다.

첫째 프로그래밍 기초 학습자를 위한 배경 설정으로 교수·학습 환경에서 프로그래밍 학습의 접근성을 높이는 데 도움을 줄 수 있을 것이다. 선행 연구를 바탕으로 기초 학습자에게 적절한 가독성 높은 코드를 제공할 수 있도록 지침을 구성하였으며 직접적인 코드 작성 전에 O/X 퀴즈, 빈칸 채우기, 파스스 문제 등 다양한 유형의 프로그래밍 연습을 제공하여 기초 학습자가 단계적으로 프로그래밍 실습을 할 수 있도록 지원하였다. 특히 교수·학습 환경에서 학습자의 개별 실습이 이루어지는 상황에서 프로그래밍 개념 학습 및 프로그래밍 연습 기능을 적용한다면 수업 과정에서 다른 개념의 이해도를 높이는 데 도움을 줄 수 있어 학생 간의 이해도 격차를 줄일 수 있을 것이다. 또한 프로그래밍 도움 기능을 통해 프로그래밍 시 겪는 오류 상황을 단계적으로 해결할 수 있도록 유도하여 학습자가 직접 문제를 해결하는 성취감을 얻는 데 도움을 줄 수 있을 것이다.

둘째 LLM을 통한 콘텐츠 생성으로 다양한 예제와 개별 학습자에게 적합한 학습 리소스를 제공할 수 있

을 것이다. 특히 학습자가 원하는 프로그래밍 언어와 개념을 활용하여 학습자 주도로 예제와 확인 퀴즈를 생성할 수 있으며 학습자의 수준이 포함된 요청으로 개별화된 연습문제와 같은 교육 콘텐츠를 제공하는데 도움을 줄 수 있을 것이다. 또한 LLM의 생성 기능을 통해 같은 개념, 같은 유형의 프로그래밍 실습 콘텐츠를 반복적으로 생성할 수 있기 때문에 기초적인 프로그래밍 학습을 위한 반복 실습에 효과적인 것으로 예상된다.

셋째 GPTs 챗봇은 다양한 교육 현장에 제한 없이 활용될 수 있을 것이다. 최적화된 상호작용 구성과 지침의 토큰 수 설정으로 교수·학습 환경에서 LLM과 많은 의사소통을 할 수 있어 다양한 콘텐츠 제공에 도움을 줄 수 있다. 또한 GPTs 플랫폼을 활용하여 추가적인 유지 보수 없이 대규모 학습 환경에서도 무리 없이 활용할 수 있을 것이다. 추가적으로 본 연구의 결과물로 공개된 지침의 수정 및 재구성은 일반적인 자연어를 통해 이루어질 수 있기 때문에 특정 교수·학습 환경에 적합한 프롬프트나 GPTs 지침으로 손쉽게 재구성되어 활용될 수 있을 것이다.

마지막으로 본 연구를 통해 구성된 프로그래밍 교육 콘텐츠 생성 챗봇은 LLM 모델의 활용 및 GPTs 플랫폼의 특성으로 다음과 같은 한계를 갖게 된다. 첫째 면밀히 제어된 지침 구성에도 LLM의 특성상 응답 및 콘텐츠가 확률적으로 생성되어 지침을 따르지 않는 응답이 간헐적으로 나타날 수 있다. 둘째 GPTs 플랫폼을 활용하여 구성되었기 때문에 OpenAI의 모델에 한해서 동작하게 되며 사용 한계 및 기타 정책이 종속되게 된다.

지능형 튜터링을 포함한 다양한 AI 및 LLM을 활용한 교육 콘텐츠의 생성은 프로그래밍 교육에서도 다방면으로 적용될 수 있을 것이다. 추후 본 연구 결과를 발전시켜 프로그래밍 교육에서의 학습자와 LLM 간의 상호작용 유형 및 각 유형에 따른 적절한 프롬프트 등과 같은 학습자 차이를 고려한 연구를 진행하고자 한다. 또한 LLM을 활용한 교수·학습 지원이 학습자의 컴퓨팅 사고력 혹은 AI 리터러시에 미치는 영향에 대한 체계적인 분석이 이루어진다면 학습의 어려움을 유발하는 다양한 요인 분석과 함께 학생들의 역량을 기르는 새로운 방향을 제시할 수 있을 것이다. 본 연구의 결과가 프로그래밍 교육 교수·학습 환경 개선을 위한 향후 연구에 활용되어 학생들의 컴퓨팅 사고력과 디지털 시대의 필수적인 역량을 기르는 데 도움을 줄 수 있기를 기대한다.

참고문헌

- [1] Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM*, 50(7), 13-18. DOI: 10.1145/1272516.1272529
- [2] Yinnan, Z., & Chaosheng, L. (2012). Training for computational thinking capability on programming language teaching. *2012 7th International Conference on Computer Science & Education (ICCSE)*, 1804-1809. DOI: 10.1109/ICCSE.2012.6295420
- [3] Weinman, N., Fox, A., & Hearst, M. A. (2021). Improving Instruction of Programming Patterns with Faded Parsons Problems. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1-4. DOI: 10.1145/3411764.3445228
- [4] Kim, S., Kim, S., Lee, M., & Kim, H. (2020). Review on artificial intelligence education for K-12 students and teachers. *The Journal of Korean Association of Computer Education*, 23(4), 1-11. DOI: 10.32431/kace.2020.23.4.001
- [5] Yau, K. W., Chai, C. S., Chiu, T. K. F., Meng, H., King, I., & Yam, Y. (2023). A phenomenographic approach on teacher conceptions of teaching Artificial Intelligence (AI) in K-12 schools. *Education and Information Technologies*, 28(1), 1041-1064. DOI: 10.1007/s10639-022-11161-x
- [6] Nayak, J., Keane, T., Linden, T., & Molnar, A. (2023). *Teaching High School Students Artificial Intelligence by Programming Chatbots*. In T. Keane & A. E. Fluck (Eds.), *Teaching Coding in K-12 Schools* (pp. 263-276). Springer International Publishing. DOI: 10.1007/978-3-031-21970-2_17
- [7] Gao, Hui & Qiu, Zhaowen & Wu, Di & Gao, Liyan. (2015). Research and Reflection on Teaching of C Programming Language Design. *Intelligent Computation in Big Data Era*, 370-377. DOI: 10.1007/978-3-662-46248-5_45.
- [8] Jeon, S., & Lee, Y. (2016). A meta analysis of programming education effects according to learning activity themes. *The Journal of Korean Association of Computer Education*, 19(2), 21-29. DOI: 10.32431/kace.2016.19.2.003
- [9] Gupta, D. (2004). What is a good first programming language? XRDS: Crossroads, *The ACM Magazine for Students*, 10(4), 7-7. DOI: 10.1145/1027313.1027320
- [10] Selby, C. C. (2015). Relationships: Computational thinking, pedagogy of programming, and Bloom's Taxonomy. *Proceedings of the Workshop in Primary and Secondary Computing Education*, 80-87. DOI:

- 10.1145/2818314.2818315
- [11] Jenkins, T. (2001). *Teaching programming—A journey from teacher to motivator*. The 2nd Annual Conference of the LSTN Center for Information and Computer Science. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9f616d5bce54f89faf9f51e45cc4197607dc1c91>
- [12] Leinonen, J., Denny, P., MacNeil, S., Sarsa, S., Bernstein, S., Kim, J., Tran, A., & Hellas, A. (2023). Comparing Code Explanations Created by Students and Large Language Models. *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 124–130. DOI: 10.1145/3587102.3588785
- [13] Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., Powell, G., Finnie-Ansley, J., & Santos, E. A. (2024). “It’s Weird That it Knows What I Want”: Usability and Interactions with Copilot for Novice Programmers. *ACM Transactions on Computer-Human Interaction*, 31(1), 1–31. DOI: 10.1145/3617367
- [14] Tian, H., Lu, W., Li, T. O., Tang, X., Cheung, S.-C., Klein, J., & Bissyand, T. F. (2023). Is ChatGPT the Ultimate Programming Assistant—How far is it? (*arXiv:2304.11938*). *arXiv*. <http://arxiv.org/abs/2304.11938>
- [15] Hellas, A., Leinonen, J., Sarsa, S., Koutchme, C., Kujanpää, L., & Sorva, J. (2023). Exploring the Responses of Large Language Models to Beginner Programmers’ Help Requests. *Proceedings of the 2023 ACM Conference on International Computing Education Research*. Chicago IL USA: ACM, 1, 93–105 DOI: 10.1145/3568813.3600139
- [16] Xiao, R., Hou, X., & Stamper, J. (2024). Exploring How Multiple Levels of GPT-Generated Programming Hints Support or Disappoint Novices. *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 1–10. DOI: 10.1145/3613905.3650937
- [17] Sarsa, S., Denny, P., Hellas, A., & Leinonen, J. (2022). Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, 27–43. DOI: 10.1145/3501385.3543957
- [18] RHEEM, J.-Y. (2011). Present State of Programming Language Education and Suggestions for Its Improvement. *Journal of Korean Institute for Practical Engineering Education*, 31(1), 56–61.
- [19] Gao, Z., Heckman, S., & Lynch, C. (2022). Who Uses Office Hours?: A Comparison of In-Person and Virtual Office Hours Utilization. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, 300–306. DOI: 10.1145/3478431.3499334
- [20] Jin, K. H., Haynie, K., & Kearns, G. (2016). Teaching Elementary Students Programming in a Physical Computing Classroom. *Proceedings of the 17th Annual Conference on Information Technology Education*, 85–90. DOI: 10.1145/2978192.2978238
- [21] Govender, I. (2009). The learning context: Influence on learning to program. *Computers & Education*, 53(4), 1218–1230. DOI: 10.1016/j.compedu.2009.06.005
- [22] Lee, M. (2012). Teaching and Learning Programming: A Constructivist Approach. *Journal of The Korean Association of Information Education*, 16(3), 363–371.
- [23] Jung, H.-W. (2019). A study on basic software education applying a step-by-step blinded programming practice. *Journal of Digital Convergence*, 17(3), 25–33. DOI: 10.14400/JDC.2019.17.3.025
- [24] Hou, X., Ericson, B. J., & Wang, X. (2024). Integrating Personalized Parsons Problems with Multi-Level Textual Explanations to Scaffold Code Writing. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education*, 2, 1686–1687. DOI: 10.1145/3626253.3635606
- [25] Zavala, L., & Mendoza, B. (2017). Precursor skills to writing code. *Journal of Computing Science in Colleges*, 32(3), 149–156.
- [26] Gomes, A., & Mendes, A. J. (2007). An environment to improve programming education. *Proceedings of the 2007 International Conference on Computer Systems and Technologies - CompSysTech '07*, 1. DOI: 10.1145/1330598.1330691
- [27] Cao, J., Li, M., Wen, M., & Cheung, S. (2023). A study on Prompt Design, Advantages and Limitations of ChatGPT for Deep Learning Program Repair(*arXiv:2304.08191*). *arXiv*. <http://arxiv.org/abs/2304.08191>
- [28] Liu, C., Bao, X., Zhang, H., Zhang, N., Hu, H., Zhang, X., & Yan, M. (2023). Improving ChatGPT Prompt for Code Generation (*arXiv:2305.08360*). *arXiv*. <http://arxiv.org/abs/2305.08360>
- [29] 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. DOI: 10.32431/KACE.2023.26.5.009
- [30] Kazemitabaar, M., Ye, R., Wang, X., Henley, A. Z., Denny, P., Craig, M., & Grossman, T. (2024). CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and

Educator Needs. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1-20. DOI: 10.1145/3613904.3642773

- [31] Lekshmi-Narayanan, A.-B., Oli, P., Chapagain, J., Hassany, M., Banjade, R., Brusilovsky, P., & Rus, V. (2024). Explaining Code Examples in Introductory Programming Courses: LLM vs Humans(arXiv:2403.05538). *arXiv*. <http://arxiv.org/abs/2403.05538>
- [32] Yetiştirilen, B., Özsoy, I., Ayerdem, M., & Tüzün, E. (2023, April 21). Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT. *arXiv*. Retrieved from <http://arxiv.org/abs/2304.10778>
- [33] *Introducing GPT-4o and more tools to ChatGPT free users*. (n.d.). Retrieved June 11, 2024, from <https://openai.com/index/gpt-4o-and-more-tools-to-chatgpt-free/>
- [34] OpenAI Platform. (2024.). Retrieved July 4, 2024, from <https://platform.openai.com>
- [35] Lawshe, C. H. (1975). A Quantitative approach to content validity. *Personnel Psychology*, 28(4), 563-575. DOI: 10.1111/j.1744-6570.1975.tb01393.x
- [36] Lee, J.-S. (2001). Delphi Research Method. Kyoyookbook.
- [37] Ayre, C., & Scally, A. J. (2014). Critical Values for Lawshe's Content Validity Ratio: Revisiting the Original Methods of Calculation. *Measurement and Evaluation in Counseling and Development*, 47(1), 79-86. DOI: 10.1177/0748175613513808
- [38] Niederberger, M., K berich, S., & members of the DeWiss Network. (2021). Coming to consensus: The Delphi technique. *European Journal of Cardiovascular Nursing*, 20(7), 692-695. DOI: 10.1093/eurjcn/zvab059



김슬기

2008년 경인교육대학교
초등교육학과(교육학사)
2016년 경인교육대학교
융합교육학과(교육학석사)
2023년 한국교원대학교
컴퓨터교육과(교육학박사)

2008년 ~ 현재 경기도 교육청 초등교사
관심분야: 정보(SW·AI) 교육, 프로그래밍 교육, 컴퓨팅사고력,
데이터 과학, 데이터 리터러시
E-Mail: tmfrlska85@gmail.com

부 록

GPTs 지침 최종안

Programming Study Tutor는 C언어와 파이썬 프로그래밍 학습을 교육적인 관점에서 지원한다.

- 개념 설명 기능을 제외한 다른 모든 기능에서는 코드를 제시하지 않는다.
- 응답 스타일과 톤은 교육자처럼 친절하고 격려하는 톤으로, 사용자에게 친근하게 다가가며, 과도한 전문 용어 사용을 피하고 쉬운 예시를 통해 설명한다.
- 코드 내어나 문제를 서술할 때 한글을 사용하며 매번 요청마다 확실하게 다른 주제와 내용과 문제를 제시한다.
- 각 언어의 주요 개념의 학습 순서는 나열된 순서와 동일

사용 방법 : 아래 사용 방법 내용을 그대로 제시

- C언어와 파이썬 프로그래밍 학습에 도움이 되는 다양한 기능을 제공합니다.
- **C언어 주요 개념 예시** : 기초 출력, 자료형, 상수와 변수, 연산자, 조건문, 반복문, 함수, 지역변수와 전역변수, 배열, 문자열, 포인터, 메모리 할당, 구조체, 공용체, 열거형
- **파이썬 주요 개념 예시** : 출력, 자료형, 연산자, 변수, 입출력, 리스트, 딕셔너리, 튜플, 조건문-if, 반복문-while, 반복문-for, 함수, 클래스와 모듈
- **주요 기능** :
 - **프로그래밍 개념 학습** : 입력된 프로그래밍 주요 개념과 이해도 확인 퀴즈를 제시합니다.
 - **프로그래밍 연습 - 코드 나열하기** : 주어진 코드를 올바른 순서로 나열하세요.
 - **프로그래밍 연습 - 빈칸이 포함된 코드 나열하기** : 주어진 코드의 빈칸을 채우고 올바른 순서로 코드를 나열하세요.
 - **프로그래밍 연습 - 문제 해결하기** : 생활 속 문제를 해결할 수 있는 코드 작성 외부 IDE를 통해 작성한 코드 입력하세요.
 - **프로그래밍 도움** : 코드 작성 중 발생한 문제를 해결하는데 도움을 받을 수 있습니다. 문제가 있는 코드를 입력하세요.
- 프로그래밍 학습을 위해 **프로그래밍 언어 - 개념 - 주요 기능**을 순서대로 입력해주세요. 예) 파이썬 - 리스트 - 코드 나열하기

1. 프로그래밍 개념 학습

- 프로그래밍 개념과 이해도 확인 퀴즈를 제시한다.
- 모든 이해도 확인 퀴즈에서 사용자가 오답을 입력하는 경우, 정답을 직접적으로 안내하지 않고 학습자가 정답을 입력할 때까지 지속적으로 힌트를 제공한다.
- 코드에서 사용되는 개념은 프로그래밍 언어 개념 순서를 따른다. 모든 예시 코드에서는 사용자가 입력한 개념과 이전 순서의 개념까지만 활용한다.
- 예시 템플릿 :
 - **개념 설명** : [생활에 가까운 소재를 활용하여 필요한 이유와 장점을 중심으로 설명하며, 핵심적인 내용만 포함한 완전한 예시 코드를 주석과 함께 제공]
 - **예시 코드 설명** : [개념을 설명하거나 예시 코드를 작성할 때에는 각 언어별로 사용자가 입력한 주요 개념 이전에 설정된 개념만 활용하여 서술]
 - **이해도 확인 퀴즈 1 (OX 퀴즈)** : [개념에 대한 OX 퀴즈를 제시]
 - **이해도 확인 퀴즈 2 (코드 빈칸 채우기)** : [개념을 활용한 간단한 빈칸이 있는 예시 코드를 제시합니다. 주요 코드 라인은 주석으로 설명을 함께 표시]
 - **이해도 확인 퀴즈 3 (코드 실행 결과)** : [개념을 활용하여 작성된 코드를 제시하면 사용자가 실행 결과를 입력하도록 함]

2. 프로그래밍 연습

- 모든 유형의 문제에 대해서 오답을 입력하더라도 절대 정답을 알려주지 않고 사용자가 정답을 입력할 때까지 힌트를 지속적으로 제공한다.

프로그래밍 연습 - 코드 나열하기

- 선택한 개념을 중심으로 Markdown 코드 양식 안에 넘버링된 코드 라인들을 올바른 순서로 나열하는 연습 문제를 아래와 같은 양식으로 제공한다.
- 모든 문제에서 개념과 관련된 잘못된 코드 한개를 하나 더 반드시 추가합니다. 잘못된 코드를 알려주지 않는다.
- 사용자는 라인별로 넘버링 된 숫자를 배열하는 것으로 응답할 수 있도록 한다.
- 예시 템플릿 :
 - **코드 상세 사항** : [코드 라인을 제시하기 전에 문제에서 제시될 코드의 상세한 흐름을 안내]
 - **코드** : [코드 나열하기 문제의 코드 조각을 나열할 때는 잘못된 코드를 포함하여 Parsons Problems의 원칙에 따라 제시]
 - **잘못된 코드를 제외하고 올바른 코드의 번호를 순서대로 입력하세요**

프로그래밍 연습 - 빈칸이 포함된 코드 나열하기 :

- 코드 나열하기와 동일한 형태에 다른 코드의 문제를 제시하며 일부 코드를 빈칸으로 설정하여 사용자가 입력하도록 한다.
- 빈칸이 포함된 라인은 주로 사용자가 입력한 프로그래밍 개념의 주요 내용을 중심으로 생성한다.

- 예시 템플릿 :
 - **코드 상세 사항** : [코드 조각을 제시하기 전에 문제에서 제시될 코드의 상세한 흐름을 안내]
 - **코드** : [일부 빈칸이 포함된 코드 조각을 포함하여 순서가 섞인 코드를 제시]
 - **빈칸이 빈칸에 알맞은 코드를 순서대로 작성하고 올바른 코드의 번호를 순서를 입력하세요**
- ### 프로그래밍 연습 - 문제 해결하기 :
 - 학생들이 흥미를 가질 수 있는 스토리텔링 형식의 문제를 제시한다.
 - 사용되는 개념은 사용자가 선택한 개념과 이전의 개념만 활용할 수 있도록 한다.
 - 사용자의 요청에도 코드는 절대 제시하지 않는다.
 - 예시 템플릿 :
 - **이이거** : [학생들이 생활에서 겪을 수 있는 문제 상황을 재미있게 이야기 형식으로 제시]
 - **문제** : [문제를 해결하기 위한 프로그래밍의 주요 방향 제시]
 - **도움말** : [문제를 해결하는데 도움이 되는 알고리즘만 제시]
- ### 3. 프로그래밍 도움: 오류가 있는 코드를 입력하도록 유도한다.
 - 사용자의 코드는 수정해서 제시 하지 않는다.
 - 모든 상황에서 힌트는 한가지의 유형만 제시한다.
 - 힌트에 대한 요청이 반복되는 경우 방향 힌트 -> 도구적 힌트를 순서대로 한번에 하나씩 제시한다.
 - 힌트를 위한 코드를 제시할 때는 사용자가 입력한 코드와 같은 부분이 하나도 없도록 제시한다.
 - 방향 힌트 추상성이 높은 힌트로, 학생들이 집중해야 할 부분을 상세히 지시
 - 도구적 힌트 오류를 해결하기 위해 무엇을 해야 하는지 상세히 설명
 - 예시 템플릿
 - 힌트 유형 : 힌트 내용