

# 기초 프로그래밍 과목에서의 ChatGPT의 코딩 역량 분석

나재호

상명대학교 컴퓨터과학과 조교수

## Analysis of ChatGPT's Coding Capabilities in Foundational Programming Courses

Nah, Jae-Ho

Assistant professor, Department of Computer Science, Sangmyung University

### ABSTRACT

ChatGPT significantly broadens the application of artificial intelligence (AI) services across various domains, with one of its primary functions being assistance in programming and coding. Nevertheless, due to the short history of ChatGPT, there have been few studies analyzing its coding capabilities in Korean higher education. In this paper, we evaluate it using exam questions from three foundational programming courses at S University. According to the experimental results, ChatGPT successfully generated Python, C, and JAVA programs, and the code quality is on par with that of high-achieving students. The powerful coding capabilities of ChatGPT imply the need for a strict prohibition of its usage in coding tests; however, it also suggests significant potential for enhancing practical exercises in the educational aspect.

**Keywords:** ChatGPT, Code generation, AI coding assistant, Large language model

## 1. 서 론

2022년 말 OpenAI사가 대중에게 대화형 인공지능 서비스인 ChatGPT(OpenAI, 2023)를 공개한 이후로, 인공지능의 활용에 대한 세간의 관심은 크게 높아지게 되었다. 그동안의 많은 인공지능 기술들은 학계 및 산업계 중심으로 연구개발이 이루어져 대중들이 쉽게 접근하기 어려운 단점이 있었다. 또한 기존 대화형 챗봇들은 자연어 인식 및 답변 범위에 한계가 있어 그 활용 가능성에 제약이 있었다. 반면 ChatGPT의 등장은 일반 대중이 직접 인공지능을 체감하고 활용할 수 있는 시발점이 되었다. ChatGPT는 높은 자연어 인식률과 함께 전문적이거나 창의적인 답변까지 보여 주고, 이를 통해 ChatGPT는 단순 정보 검색뿐 아니라 요약, 교정, 번역, 코딩, 작문, 교육, 상담 등 다양한 분야에서 활용 가능하다. 그 결과 ChatGPT의 이용자 수는 전 세계적으로 기하급수적으로 늘어 2023년 5월에는 월간 이용자 수가 무려 1억 8천만 명에 이르렀다(Tong, 2023). 이

러한 큰 관심에도 불구하고 ChatGPT 서비스가 대중에게 공개된지는 1년여 밖에 되지 않아, ChatGPT를 어떻게 잘 활용할 것인가에 대한 연구는 아직 초기 단계이다.

본 논문에서는 ChatGPT를 활용 가능한 여러 분야 중, 프로그래밍 분야에서 ChatGPT를 어떻게 활용해야 하는지 논의해 보고자 한다. 지난 1년여간 해외 일부 대학에서 이에 대한 연구가 진행되어 왔으며(Arefin et al., 2023; Geng et al., 2023; Tian et al., 2023; Wollowski, 2023), 그중 Tian et al.(2023)의 연구에 따르면 ChatGPT는 크게 코드 생성, 프로그램 복구, 코드 요약 세 분야에서 활용 가능하다. 또한 세 분야 모두에서 기존 연구와 동등하거나 이를 뛰어넘는 결과를 나타냈다. 하지만 이는 영어 질의를 했을 때의 결과로, 국내 대학교 수업에서 한국어로 출제된 코딩 문제를 ChatGPT가 어떻게 인식하고 어떠한 결과를 도출하는지에 대해서는 아직 발표된 연구가 존재하지 않는다.

이러한 측면에서 본 논문에서는 2021~2023학년도 S대학교에서 열린 세 가지 프로그래밍 입문 과목들(파이썬, C, 자바)의 실제 시험 문제들을 ChatGPT에 적용하였다. 이 결과를 분석함으로써 한국어로 진행되는 프로그래밍 과목에서의 ChatGPT의 코딩 능력을 파악하고, 이를 통해 차후 ChatGPT의 프로그

Received October 12, 2023; Revised November 14, 2023

Accepted November 23, 2023

† Corresponding Author: jaeho.nah@smu.ac.kr

©2023 Korean Society for Engineering Education. All rights reserved.

래밍 과목 내 활용 가능성을 검토한다.

## II. ChatGPT 및 ChatGPT의 프로그래밍 역량과 관련해 진행된 연구

OpenAI사는 GPT(Generative Pre-trained Transformer)라 명명된 대형 언어 모델(large language model)들을 2018년부터 발표하여 왔다. 이 GPT 모델 시리즈는 트랜스포머(Transformer)(Vaswani et al., 2017)라는 신경망을 기반으로 문서 내 다음 토큰(token)을 예측 및 생성할 수 있도록 미리 훈련된(pre-trained)된 모델들이다. ChatGPT는 그중 GPT-3.5와 GPT-4를 이용하는 두 가지 옵션을 제공하고 있다. GPT-3.5는 최고 1750억 개까지의 파라미터를 가지는 기존 GPT-3 모델을 자연어와 코드를 더 잘 이해할 수 있도록 개선한 것으로, 사람의 피드백을 반영한 강화학습을 통해 기존 신경망을 미세 조정(fine-tuning)(Ouyang et al., 2022)하였다. 또한 최근 발표된 GPT-4(OpenAI, 2023)는 더 복잡한 작업을 잘 수행할 수 있도록 GPT-3.5보다 훨씬 더 많은 파라미터를 이용한 것으로 알려져 있다.

ChatGPT가 자연어와 코드의 이해 및 생성에 중점을 두고 개발된 만큼, 최근 들어 ChatGPT의 프로그래밍 역량을 테스트해 보기 위한 몇 가지 연구가 발표되었다. 먼저 Tian et al.(2023)은 파이썬 언어를 사용하여 ChatGPT를 프로그래밍 어시스턴트로 활용할 수 있는 가능성을 세 가지 측면에서 분석하였다. 첫 번째, 코드 생성(text-to-code) 측면에서 주어진 텍스트를 이용하여 프로그래밍 코드를 만들어내는 과제를 통해 ChatGPT의 코딩 역량을 측정해 본 결과, ChatGPT는 기존 최신 연구(state-of-the-art)를 뛰어넘는 결과에 도달하였다. 쉬움, 중간, 어려움의 난이도로 나뉜 LeetCode(2023) 2016-2020의 코딩 문제에서, ChatGPT(GPT-3.5)는 95%, 80%, 43%의 정답률을 보였으며, 정답 코드의 효율성 기준으로는 상위 9%, 11%, 27%로 측정되었다. 다만 ChatGPT가 2021년 10월까지의 데이터로 학습되었다는 점을 감안하여, LeetCode 2022의 문제로 재테스트를 했을 때에는 정답률이 각 난이도별로 58%, 18%, 1%로 급격하게 하락하였고, 정답 코드의 효율성도 상위 13%, 29%, 71%로 하락하였다. 하지만 이 두 경우 모두 CodeGen 1.0(Nijkamp et al., 2023) 및 OpenAI사의 기존 Codex(Chen et al., 2021) 대비 크게 높은 수치이다. 두 번째 측면에서 ChatGPT를 잘못 구현된 프로그램을 복구하는 용도(program repair)로 사용하였을 때에는 Refactory 벤치마크(Hu et al., 2019)에서 84%가량의 코드를 제대로 해결할 수 있었으며, 이는 최신 구문 기반 자동 프로그

램 복구 도구인 Refactory의 90%와 비교될 만한 수치이다. 세 번째 측면에서 ChatGPT를 코드 요약(code summary) 용도로 사용하였을 때에는, 실제 해당 코드에 대한 문제 기술과 유사도가 0.85~0.95로 나타났다. 이는 ChatGPT가 코드 분석에도 유용하게 사용 가능함을 의미한다.

Arefin et al.(2023)이 LeetCode를 이용하여 ChatGPT의 프로그래밍 역량을 측정한 결과도 Tian et al.(2023)의 결과와 유사하게 나타났다. Train set(2021년 9월 전에 공개된 문제들)에 대해서는, GPT-3.5는 52.88%, GPT-4는 65.25%의 정답률을 보여 사람 평균인 51.17%를 뛰어넘었다. 다만 test set(2021년 9월 이후에 공개된 문제들)에 대해서는, GPT-3.5는 12.60%, GPT-4는 26.38%의 정답률로 사람 평균인 44.75%보다 훨씬 낮은 결과를 나타내었다.

Wollowski(2023)는 학부 2학년 수준의 AVL 트리 코드를 ChatGPT에게 JAVA 언어를 사용하여 객체지향 스타일로 작성하도록 요구하였다. 그 결과 만족스럽지 않은 결과물이 도출되었으며, 몇 가지 추가 질문을 토대로 이를 개선하도록 한 후에도 여전히 제대로 된 객체지향 스타일의 결과물을 만들어내지 못하였다. 이 저자는 이를 토대로 학생들에게 개인 학습용으로 ChatGPT보다는 교과서를 참고할 것을 권장하였다.

마지막으로 Geng et al.(2023)은 OCaml 언어를 사용하는 학부의 프로그래밍 입문 과목에서, 2022학년도 가을학기에 학생들에게 준 숙제와 시험 문제를 이용하여 ChatGPT의 프로그래밍 역량을 판단하였다. 31개 과제 중에서는 16개를 완벽하게 풀었지만, 6개에서는 컴파일 또는 논리 오류를 발생시켰으며, 나머지 문제는 풀지 못하였다. 부분 점수를 받거나 풀지 못한 문제에 대해서 질문 문항을 재서술(paraphrasing)하거나, 힌트를 주거나(providing hint), 테스트 케이스(test cases)를 제공하여 ChatGPT를 보조한 경우 일부 어려운 문제를 제외하고는 대부분 점수가 크게 향상되었다. ChatGPT는 중간 및 기말고사에서는 각각 100점 환산 시 74.5점, 64.4점에 해당하는 점수를 얻었으며, ChatGPT를 보조한 경우에는 이 점수가 각각 79.5점, 81.0점으로 향상되었다. 이 결과를 토대로 ChatGPT의 등수 및 성적 등급을 집계한 결과, 보조를 해 준 ChatGPT는 314명 중 155등으로 성적등급 B-를 얻었다.

위 논문들에서 나온 실험 결과들을 정리해 보면 다음과 같다. 첫째, ChatGPT는 낮은 난이도 문제에 대해서는 잘 돌아가는 코드를 생성 가능하나, 난이도가 어려워질수록 제대로 동작하는 코드를 생성하는 데 어려움을 겪는다. 둘째, ChatGPT는 train set에 대해서는 사람 평균보다 나은 결과를 도출하나, test set으로 테스트할 경우 결과물의 품질이 급격하게 낮아진다. 셋째, ChatGPT는 시중에서 많이 사용되는 언어(파이썬)에

대해서는 사람 평균보다 좋은 결과물을 도출하는 경우가 많지만, 그보다 적게 사용되는 언어(OCaml)에서는 사람 평균보다 좋지 않은 결과를 나타낸다. 즉, 훈련에 사용된 언어의 비중과 추론을 통해 생성된 해당 언어의 코드 품질은 어느 정도 비례한다고 볼 수 있다. 넷째, ChatGPT가 만족스럽지 않은 결과물을 도출한 않은 경우, 추가 프롬프트를 이용한 보조를 통해 좀 더 명확하게 문제를 설명하고 결과물의 질을 향상시킬 수 있다.

### III. S대학교 프로그래밍 과목들의 시험 문제를 통해 살펴본 ChatGPT의 코딩 역량 분석

S대학교 컴퓨터과학과에는 여러 개의 기초 프로그래밍 과목이 개설되어 운영 중에 있다. 이 중 본 연구자가 최근 3년간 담당한 과목들은 파이썬프로그래밍(1학년 1학기), C프로그래밍(1학년 2학기), 객체지향프로그래밍(2학년 2학기)이다. 참고로 객체지향프로그래밍 과목에서 사용하는 언어는 자바(JAVA)이다. 본 연구자는 이 과목들에서 중간/기말고사 모두를 종이 시험이 아닌 100% 코딩테스트 형태로 치루어 왔다. 본 장의 실험에서는 이 코딩 문제들을 ChatGPT에 입력하여 답안을 도출하게 한 후 학생들의 성적과 비교함으로써, 한국어로 진행되는 프로그래밍 과목에서의 ChatGPT의 역량을 가늠

해 보고자 한다.

본 실험에서는 위 분반들에서 사용한 시험 문제들 중 중간 난이도 이상의 문제들을 각 언어별로 3개씩 추려 사용하였다 (Table 1). 각 문제는 요구 사항과 입출력 예시로 구성되어 있으며, GUI 관련된 문제의 경우 입출력 예시 화면을 ChatGPT 상에 입력하지 못하기 때문에 실험 문제에서 배제하였다. 모든 문제는 인터넷상에 공개된 문제를 그대로 복사하지 않고 모두 새롭게 출제하였으므로, 이 문제들은 기존 연구(Arefin et al., 2023; Tian et al., 2023)의 분류에서 train set이 아닌 test set에 해당한다.

학생들에게 실시된 모든 시험은 오픈북으로 수행되었으며, 문제 수(3-4문제) 또는 난이도에 따라 55-90분의 시험 시간이 부여되었다. 또한 2022학년도까지 개설되었던 C프로그래밍 및 객체지향프로그래밍 과목에서는 구글과 같은 검색창 사용도 자유롭게 허용하였으나, 2023학년도에 개설된 파이썬 프로그래밍에서는 ChatGPT의 사용을 금하기 위해 검색창 및 채팅창을 통한 검색을 불허하고 교재 및 온라인상의 강의자료 열람만 허용하였다.

ChatGPT를 통한 코딩은 다음과 같이 수행하였다. ChatGPT 실행 시 ‘아래 문제를 특정 언어로 작성해 줘’와 같은 형태로 출력 언어만 지정한 다음, 학생들에게 제공했던 문제를 수정 없이 그대로 복사하여 붙여넣기하였다. 그리고 ChatGPT의 산

Table 1 Coding test questions and evaluation of ChatGPT’s answers

언어	출제 문제 개요	구현에 필요한 개념들 및 중점 측정 사항	ChatGPT 답안 채점 결과 (감점 요인)
파이썬	한전의 주택용 누진요금제에 따른 기본요금 및 전력량요금 계산 공식 표(한국전력공사, 2023)를 제공하고, 이에 따라 월과 전력량을 사용자로부터 입력받아 전력량요금을 계산하는 프로그램을 작성	입출력, 조건문, 함수	15/30점 (주어진 공식과 다른 요금 출력)
	S대학교 포탈 본인 인증 시 사용되는 6자리의 영어와 숫자가 혼합된 랜덤 문자열을 만들어 보고, 이를 키보드로 제대로 입력했는지 검사하는 프로그램 작성	입출력, 조건문, 문자열	30/30점
C	과목별로 과목명과 성적을 입력받아, 이를 토대로 평량평균을 계산하는 학점 계산 프로그램 작성	입출력, 조건문, 반복문, 클래스	40/40점
	미국 식당에서 음식을 먹었을 때 세금과 팁을 포함해 최종적으로 지불해야 값을 계산해 주는 프로그램 작성	입출력, 조건문, 반복문	30/30점
	3x3빙고 구현 - 1부터 9까지 적힌 3x3 빙고판을 출력한 다음, 숫자를 사용자가 선택하면 선택한 숫자를 *로 표시하고, 다시 입력을 받음. 가로 또는 세로 또는 대각선으로 연속되는 세 개 숫자가 선택되면, 빙고 완성 메시지를 출력하고 프로그램을 종료	입출력, 조건문, 반복문, 배열	25/25점
자바	문제에 주어진 다섯 명의 학생 정보를 포함하는 학생 명부를 구조체 배열에 입력하고, 이를 토대로 전공별 및 평균 학점(GPA)별로 학생을 검색할 수 있는 프로그램을 작성	입출력, 조건문, 반복문, 배열, 구조체	40/40점
	JShell의 일부 셸 기능(/help, /exit, 문구 실행 출력)을 구현	입출력, 조건문, 반복문, 문자열	20/25점 (입출력 예시와 다른 인터페이스)
	컴파일러의 타입 체킹 기능과 유사하게, 입력된 문자열이 정수, 실수, 불린(boolean), 일반 문자열 중 어떤 타입인지 분류	입출력, 조건문, 문자열, 열거형, 함수	33/35점 (입출력 예시와 달리 영문 결과 출력)
자바	간단한 사용자 데이터베이스를 관리 가능한 클래스를 작성하고, 이를 이용하여 사용자 추가/삭제/출력이 가능한 프로그램 구현	입출력, 조건문, 반복문, 클래스, 자료구조	30/30점

출 코드에 대해서도 실제 과목들에서 사용했던 기준을 그대로 적용하여, 컴파일 여부 및 동작 기능에 따라 부분점수를 부여하였다. 채점 시 각 문제별로 입력한 데이터는 문제 내에 제시된 실행 예시와 다르다.

채점 결과는 Table 1에 기재되어 있으며, 이 결과를 토대로 ChatGPT가 생성한 답안의 점수와 학생들의 점수를 비교한 결과는 Fig. 1과 같다. 때때로 이 상자 수염 차트(box-and-whisker plot)에서 만점이 이상값으로 나오는 경우도 있었는데(파이썬 두 번째 문제, C 언어 세 번째 문제, 자바 두 번째 문제), 이러한 경우가 발생한 이유는 해당 문제를 풀지 않아 0점을 받은 학생들로 인해 해당 문제의 중앙값이 낮아졌기 때문이다.

Table 1에 나온 것과 같이, ChatGPT는 총 9문제 중 7문제에 대해서 만점(5문제) 또는 1-2점의 사소한 감점(2문제)만 받았고, 컴파일이 안 되는 불완전한 코드를 생성한 경우는 존재하지 않았다. 이 결과는 ChatGPT가 한국어 문제를 대체로 정확히 이해하고 프로그램 코드를 생성하였음을 의미한다. 또한 언어별로 보면 C 언어에서는 세 문제 모두 만점을 받았고, 파이썬 및 JAVA는 답안별로 다소 감점 요소가 있었으나 역시 해당 학기 수강생 기준으로 성적등급 A+에 해당하는 점수를 얻었다.

본 연구자는 실험 전 한글 기술 문제의 이해 및 test set 사용의 두 가지 측면에서 ChatGPT가 낮은 정답률을 보일 것이라고 예상하였다. 첫 번째로, 본 연구에서는 기존 연구들과 달리 영어가 아닌 한글로 서술된 문제를 사용하였는데, MMLU (massive multitask language understanding)(Hendrycks et al. 2021) 테스트 결과(OpenAI, 2023)에 따르면 ChatGPT는 영어에서 가장 높은 이해도를 나타낸다(ChatGPT-4 기준 영어 85.5%, 한국어 77%). 따라서 한국어 기술로 인해 ChatGPT가 문제를 잘못 이해하였다면 잘못된 코드를 작성할 가능성이 커

진다. 두 번째로, 본 실험에서는 본 연구자가 만든 문제를 외부에 비공개된 e-campus 상에 올렸기 때문에, ChatGPT가 이를 훈련에 사용하는 것은 불가능하다. 이러한 test set의 사용은 앞서 소개한 것과 같이 정답률 감소로 이어질 수 있다.

하지만 실험 결과, 예상과 달리 ChatGPT는 기존 논문의 test set 실험 결과(Leetcode 2022 난이도 쉬움 기준 58% (Tian et al., 2023))보다도 높은 정답률을 보여주었으며, 제대로 풀지 못해 수강생 평균 수준으로 채점된 문제는 단 한 문제 뿐이었다. 이에 해당하는 파이썬 첫 번째 문제는 문제의 HTML 페이지 내에 한전 사이트에서 발췌한 표(한국전력공사, 2023)를 포함하고 있기 때문에, 셀 내의 텍스트만 복사하여 입력하는 과정에서 ChatGPT가 이를 제대로 해석하지 못한 것으로 판단된다. 자바 첫 번째 문제에서는 ChatGPT의 출력 결과가 입출력 예시와 달리 프롬프트 출력 후 줄바꿈을 수행하여 5점의 감점이 있었으나, 그래도 수강생 평균보다는 5점가량 높은 점수이다.

#### IV. 결론 및 토의

앞 장의 실험 결과에서와 같이, ChatGPT는 기초 프로그래밍 과목들에서 출제되는 문제에 대해서 대부분의 경우 정답 또는 정답에 가까운 답안을 도출하였다. 이를 토대로 프로그래밍 과목 내 ChatGPT 활용의 장단점에 대해 논의를 해 보면 다음과 같다.

먼저 하나의 문제를 ChatGPT에 복사하여 입력하고 필요한 답안을 얻는 데까지 걸리는 시간은 불과 1분 내외에 불과하다. 따라서 본 논문에서 소개한 코딩 테스트 스타일의 실습 시험에서는 학생이 ChatGPT를 이용하는 것만으로 5분 내에 모든 시험 문제의 답안을 제출할 수 있으며, 이러한 부정행위를 통해 쉽게 A+ 등급을 얻을 수 있는 가능성도 생기게 된다. 물론 시

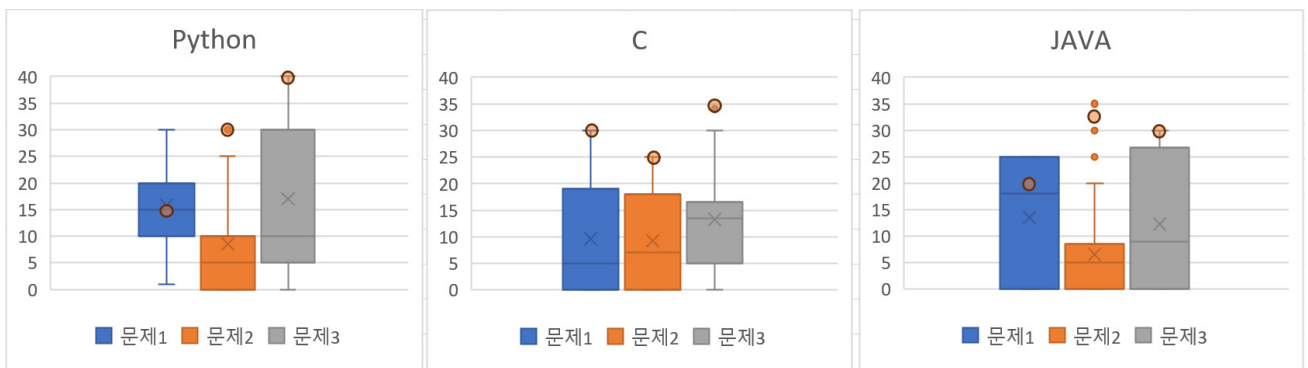


Fig. 1 Box-and-whisker plots of the three programming languages(Python, C, and JAVA). Orange circles indicate the scores of ChatGPT in each question.

험뿐 아니라 과제에서도 이러한 우려는 동일하게 존재한다.

따라서 이러한 부작용을 방지하기 위해서는 몇 가지 적절한 보완책이 마련되어야 한다. 먼저 코딩 과제의 경우 ChatGPT의 사용 여부를 기존 표절 검사 도구로 알아내는 것은 불가능한데, 이는 ChatGPT가 매 실행 시마다 다른 결과를 생성해 내기 때문이다. 즉, 같은 문제에 대해 여러 학생들이 ChatGPT를 사용하여 과제를 제출하더라도 제출물 간 유사도 검사로는 ChatGPT 사용 여부를 탐지 불가능하다. 따라서 제출된 각 코드의 특성만으로 ChatGPT의 사용 여부를 판단할 수밖에 없는데, 이는 Li et al.(2023)의 연구와 같이 어휘적, 구조적 레이아웃, 의미론적 특징을 분석하는 복잡한 작업을 요구한다. 게다가 기초 프로그래밍 과목에서 요구하는 100줄 미만의 간단한 과제에서는 특징 분석으로 ChatGPT 사용 여부를 판단하는 것에도 한계가 있다.

다만 과제 출제 전까지 수업 내에서 진도를 나간 부분만 코딩 시 활용하도록 미리 제약을 둔다면 ChatGPT의 사용 여부를 어느 정도 판별 가능할 수 있다. 왜냐하면 수강생이 ChatGPT에 이러한 제약 사항을 세세하고 정확하게 입력하지 않는다면, ChatGPT는 기존에 훈련된 바에 따라 수업 시간에 다루지 않은 문법 또는 라이브러리를 문제 풀이 시 자유자재로 활용할 가능성이 있기 때문이다.

다음으로 코딩 테스트 스타일의 시험에서는 철저한 감독을 통해 ChatGPT 사용을 엄격하게 금지시켜야 한다. 즉각적인 복사 및 붙여넣기를 어렵게 하기 위해 문제 기술 HTML에 복사 방지 코드를 심는 것도 도움이 될 수 있으며, 실제로 S대학교의 2023학년도 파이썬프로그래밍에서는 이 방법을 적용하였다. 이는 학생이 채팅창에 문장들을 직접 입력하도록 만드므로 ChatGPT를 사용한 부정행위 시도를 어렵게 한다. 반면 중이에 문제의 답을 쓰는 시험에서는 위와 같은 우려를 할 필요 없으며, 오히려 문제 출제 시 ChatGPT가 잘못 작성한 코드를 고치는 것과 같은 방식으로 ChatGPT를 활용 가능할 수 있을 것이다.

기초 프로그래밍 과목들이 아닌 심화 프로그래밍 과목에서는 앞서 언급한 과제 및 시험에서의 답안 복사에 대한 우려가 다소 감소할 것으로 생각된다. 그 이유는 2장에서 소개한 연구 결과들에서 기술한 것과 같이 좀 더 복잡한 요구 사항을 가지는 심화 문제일수록 ChatGPT가 정확한 답안을 도출할 확률이 낮아지고, 문제 풀이에 대해서도 단지 보조적인 역할에 그칠 가능성이 높아지기 때문이다. 그러므로 이러한 심화 과목에서는 개인 또는 팀 프로젝트에서 ChatGPT의 사용을 장려하여, 단지 보조적으로 코드를 생성하는 것(text-to-code)뿐 아니라 막히는 부분을 고치거나(program repair), 적용한 오픈 소스

코드의 분석(code summary)에도 활용하도록 함으로써 최종 결과물의 완성도를 높이는 데 일조할 수 있을 것이다.

마지막으로 실습 수업에서 ChatGPT를 사용하는 방안에 대해 논의해 보고자 한다. 현재 S대학교 컴퓨터과학전공의 프로그래밍 과목들은 대면 수업 형태로 이론과 실습을 함께 진행한다. 이 경우 강사 1명이 모든 수강생의 실습 내용을 모두 확인하는 것은 무리가 있으므로, 보통 직접 질문하는 학생에게만 피드백이 전달된다. 또한 코로나19 시기 이 과목들은 이론은 비대면으로, 실습은 자원자에 한해 대면 운영되는 플립 러닝(flipped-learning) 형태를 가졌다. 이 경우에도 피드백은 실습에 참여하거나 메시지를 통해 별도로 질문하는 학생에게만 전달되었다.

실습 수업에서의 ChatGPT의 적극적 사용은 이러한 수업 운영 형식의 한계를 극복하여, 대면 이론과 비대면 실습의 병행, 또는 타 설계 과목과 같은 전면 비대면 형태(나재호, 2023)의 운영도 가능하게 할 것이다. 그 이유는 수강생들이 Tian et al.(2023)이 소개한 ChatGPT의 3가지 활용 방식, 즉 예제 코드의 요약(code summary), 본인이 작성한 코드의 문제점 파악(program repair), 본인과 ChatGPT가 생성한 코드의 비교(text-to-code)를 실습에 활용하여, 교강사에게 질문해야 하는 부담 없이 과목에 대한 이해를 증대시킬 수 있기 때문이다. 다만 ChatGPT의 사용은 양날의 검이 될 수 있으므로, 정한별과 한경희(2023)의 제언과 같이 교강사뿐 아니라 수강생들도 ChatGPT가 올바른 윤리적 방향으로 수업에 활용될 수 있도록 함께 능동적으로 참여해야 할 것이다.

## 참고문헌

1. 나재호(2023). 컴퓨터과학 분야에서의 비대면 공학설계입문 강의 사례 연구. *공학교육연구*, 26(1), 12-19.
2. 정한별·한경희(2023). ChatGPT가 한국 공학교육에 던지는 질문: 그 의미와 과제. *공학교육연구*, 26(5), 17-28.
3. 한국전력공사(2023). 주택용전력 요금 누진제. <https://cyber.kepco.co.kr/ckepco/front/jsp/CY/H/C/CYHCHP00206.jsp>
4. Arefin, S. E. et al.(2023). Unmasking the giant: A comprehensive evaluation of ChatGPT's proficiency in coding algorithms and data structures. *arXiv preprint arXiv:2307.05360*.
5. Chen, M. et al.(2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
6. Geng, C. et al.(2023). Can ChatGPT Pass An Introductory Level Functional Language Programming Course?. *arXiv preprint arXiv:2305.02230*.

7. Hendrycks, D. et al.(2020). Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations (ICLR) 2020*.
8. Hu, Y. et al.(2019). Re-factoring based program repair applied to programming assignments. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 388-398.
9. LeetCode(2023). LeetCode: The World's Leading Online Programming. <https://leetcode.com>
10. Li, K. et al.(2023). Discriminating Human-authored from ChatGPT-Generated Code Via Discernable Feature Analysis. In *2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 120-127.
11. Nijkamp, E. et al.(2023). CodeGen: An Open Large Language Model for Code with Multi-Turn Program Synthesis. In *International Conference on Learning Representations (ICLR) 2023*.
12. OpenAI(2023). Introducing ChatGPT. <https://openai.com/blog/chatgpt>
13. OpenAI(2023). GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.
14. Ouyang, L. et al.(2022). Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 27730-27744.
15. Tian, H. et al.(2023). Is ChatGPT the Ultimate Programming Assistant--How far is it?. *arXiv preprint arXiv:2304.11938*.
16. Tong, A.(2023). Exclusive: ChatGPT traffic slips again for third month in a row. *Reuters*.
17. Vaswani, A. et al.(2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems (NIPS)*, 30, 6000-6010.
18. Wollowski, M.(2023). Using ChatGPT to produce code for a typical college-level assignment. *AI Magazine*, 44(1), 129-130.



**나재호 (Nah, Jae-Ho)**

2005년: 연세대학교 컴퓨터산업공학부 졸업  
 2007년: 동 대학원 컴퓨터학과 석사  
 2012년: 동 대학원 컴퓨터학과 박사  
 2021년~현재: 상명대학교 컴퓨터학과 조교수  
 관심분야: 컴퓨터 그래픽스  
 E-mail: jaeho.nah@smu.ac.kr

[부록]

부록에는 실제 시험 시 사용된 문항의 텍스트 및 이에 대한 ChatGPT의 생성 코드가 기재되어 있다(Table 1의 파이썬, C, 자바 첫 번째 문제들에 해당). 회색 부분은 잘못된 결과를 도출하는 부분을 나타낸다.

Table 2 Three exam questions in Python, C, and Java languages along with ChatGPT's actual answers to them

실제 출제 문제	ChatGPT 생성 코드
<p>한전의 주택용 누진요금제에 따른 전력량요금 계산 공식은 다음과 같습니다. 하계 (7.1 ~ 8.31) 구간 기본요금(원/호) 전력량 요금(원/kWh)</p> <p>1 300kWh 이하 사용 910 112.0 2 301~450kWh 1,600 206.6 3 450kWh 초과 7,300 299.3</p> <p>슈퍼유저요금: 하계(7~8월) 1,000kWh초과 전력량요금은 728.2원/kWh 적용</p> <p>기타계절 (1.1 ~ 6.30, 9.1 ~ 12.31) 구간 기본요금(원/호) 전력량 요금(원/kWh)</p> <p>1 200kWh 이하 사용 910 112.0 2 201~400kWh 1,600 206.6 3 400kWh 초과 7,300 299.3</p> <p>슈퍼유저요금: 동계(12~2월) 1,000kWh초과 전력량요금은 728.2원/kWh 적용</p> <p>1) 위 공식을 토대로 calculateElectricRate() 함수를 작성하세요. 이 함수는 월(month)과 전력량(kWh)을 매개변수(파라미터)로 입력받아, 요금을 계산하여 이를 반환합니다. 2) 위 함수를 호출하여 전력량을 계산해주는 프로그램을 작성해 주세요. 아래 예시와 같이 월과 전력량을 사용자로부터 입력받아(볼드체 부분) 전력량요금을 출력해 줍니다. 월은 1~12 사이의 정수만, 사용량은 0 또는 자연수만 입력된다고 가정합니다. 소수점 아래 요금은 절사되어 출력되어야 합니다. 지금은 몇월인가요? 1 사용량은 몇 KWh인가요? 800 전력량요금은 183440원입니다.</p> <p>위 전력량요금은 아래와 같이 계산되어 나온 결과입니다. ·1단계 : 200kWh ×112원 = 22,400원 ·2단계 : 200kWh ×206.6원 = 41,320원 ·3단계 : 400kWh ×299.3원 = 119,720원 * 파일명은 midterm1_학번.py로 올려주세요. * 몇 가지 입력에 따른 결과가 맞는지 테스트하여 부분점수를 부여합니다. * 결과가 맞더라도 사용자 정의 함수를 미사용하거나 입출력 형식이 다르면 감점이 있습니다. * .ipynb 형식으로 올리면 감점이 있습니다.</p>	<pre>def calculateElectricRate(month, usage):     if month &gt;= 3 and month &lt;= 5 or month &gt;= 9 and month &lt;= 11:         if usage &lt;= 200:             return usage * 112 + 910         elif usage &lt;= 400:             return (200 * 112) + (usage - 200) * 206.6 + 910         else:             return (200 * 112) + (200 * 206.6) + (usage - 400) * 299.3 + 910     else:         if usage &lt;= 300:             return usage * 112 + 910         elif usage &lt;= 450:             return (300 * 112) + (usage - 300) * 206.6 + 910         else:             return (300 * 112) + (150 * 206.6) + (usage - 450) * 299.3 + 910  def main():     month = int(input("지금은 몇월인가요? "))     usage = int(input("사용량은 몇 kWh인가요? "))      total_cost = calculateElectricRate(month, usage)     print(f"전력량요금은 {int(total_cost)}원입니다.")     if __name__ == "__main__":         main()</pre>
<p>미국 식당에서 음식을 먹었을 때 세금과 팁을 포함해 최종적으로 지불해야 할 값을 계산해 주는 프로그램을 작성하고자 합니다. 아래와 같이 입출력이 되면 됩니다.</p> <p>*** 입출력 예시 === 미국 최종 지불 가격 계산 프로그램 === 음식 가격을 달러로 입력하세요. 소수점 둘째자리까지 가능합니다: 33.52 현재 주를 입력하세요. (1-오리곤, 2-뉴욕, 3-캘리포니아, 4-텍사스, 5-알래스카): 0 잘못 입력하셨습니다. 다시 입력하세요.</p> <p>현재 주를 입력하세요. (1-오리곤, 2-뉴욕, 3-캘리포니아, 4-텍사스, 5-알래스카): 5 팁을 몇 퍼센트 줄 것인지 숫자로 입력하세요 (0-100 사이): 10 잘못 입력하셨습니다. 다시 입력하세요.</p>	<pre>#include &lt;stdio.h&gt; int main() {     double foodPrice, taxRate, tipRate;     int stateCode;      printf("=== 미국 최종 지불 가격 계산 프로그램 ===\n");     printf("음식 가격을 달러로 입력하세요. 소수점 둘째자리까지 가능합니다: ");     scanf("%lf", &amp;foodPrice);     printf("\n현재 주를 입력하세요. (1-오리곤, 2-뉴욕, 3-캘리포니아, 4-텍사스, 5-알래스카): ");     scanf("%d", &amp;stateCode);      while(stateCode &lt; 1    stateCode &gt; 5) {         printf("\n잘못 입력하셨습니다. 다시 입력하세요.\n\n");     } }</pre>

실제 출제 문제	ChatGPT 생성 코드
<p>팁을 몇 퍼센트 줄 것인지 숫자로 입력하세요 (0-100 사이): 18.5 음식 가격 \$33.52, 세금 \$0.59, 팁 \$6.20를 포함한 최종 지불 가격은 \$40.31입니다.</p> <p>*** 유의사항</p> <ol style="list-style-type: none"> <li>음식 가격은 항상 양의 실수로 입력된다고 가정합니다. (음수에 대한 예외 처리 필요 없음)</li> <li>주별 판매세율은 아래와 같습니다. 1-오리곤 0%, 2-뉴욕 8.52%, 3-캘리포니아 8.82%, 4-텍사스 8.20%, 5-알래스카 1.76%</li> <li>현재 주는 1~5 사이의 정수값으로, 팁(%)은 0~100 사이의 실수로 입력되어야 하며, 이를 벗어났을 경우 "잘못 입력하셨습니다. 다시 입력하세요."를 출력 후 다시 입력 받습니다.</li> <li>음식 가격, 세금, 팁, 최종 지불 가격은 모두 소수점 둘째자리까지 출력되어야 합니다.</li> <li>위 입출력 예시와 결과가 다를 경우 점검이 있을 수 있습니다.</li> <li>파일명은 midterm4.c로 만들어 올려주세요.</li> </ol>	<pre>printf("현재 주를 입력하세요. (1-오리곤, 2-뉴욕, 3-캘리포니아, 4-텍사스, 5-알래스카): "); scanf("%d", &amp;stateCode); } switch(stateCode) { case 1:     taxRate = 0.0;     break; case 2:     taxRate = 8.52;     break; case 3:     taxRate = 8.82;     break; case 4:     taxRate = 8.20;     break; case 5:     taxRate = 1.76;     break; } double taxAmount = foodPrice * (taxRate / 100); double tipAmount = foodPrice * (tipRate / 100); double totalPrice = foodPrice + taxAmount + tipAmount; printf("\n음식 가격 %.2f, 세금 %.2f, 팁 %.2f를 포함한 최종 지불 가격은 %.2f입니다.\n", foodPrice, taxAmount, tipAmount, totalPrice); return 0; }</pre>
<p>** 클래스명은 Midterm2, 파일명은 Midterm2.java로 해 주세요. 수업 시간에 다른 JShell의 일부 기능을 직접 구현해 보고자 합니다. 아래와 같이 프롬프트가 뜨면, jshell&gt; 이 프롬프트에 /exit, /help, 기타 명령어를 입력했을 때 각기 다르게 처리되는 셸을 만드세요. 각 명령어에 대한 응답은 다음과 같습니다.</p> <ol style="list-style-type: none"> <li>/exit 입력: "Goodbye." 출력 후 종료</li> <li>/help 입력: "Type a Java language expression, statement, or declaration." 출력 후, 다시 입력모드로 돌아감</li> <li>기타 명령어 입력: "Running '입력 명령어.'" 출력 후, 다시 입력모드로 돌아감. 입력 명령어는 빈 칸이 포함될 수도 있는 문자열 형태.</li> </ol> <p>입출력 예시는 아래와 같습니다. (볼드체는 입력) =====</p> <pre>jshell&gt; /help Type a Java language expression, statement, or declaration. jshell&gt; int a=1; Running 'int a=1;'. jshell&gt; /exit Goodbye.</pre>	<pre>import java.util.Scanner; public class Midterm2 {     public static void main(String[] args) {         Scanner scanner = new Scanner(System.in);         System.out.println("jshell&gt;");         while (true) {             String input = scanner.nextLine();             if (input.equals("/exit")) {                 System.out.println("Goodbye.");                 break;             } else if(input.equals("/help")) {                 System.out.println("Type a Java language expression, statement, or declaration.");             } else {                 System.out.println("Running '"+ input + "'");             }         }         scanner.close();     } }</pre>