基于提示工程，使用 ChatGPT 生成编程考试题与答案

Jongwook Si 和 Sungyoung Kim

摘要—在计算机科学领域，鼓励学生学习 Python、C语言 ++ 和 Java 等多种编程语言，帮助他们掌握广泛的技术技能并提升解决问题的能力。然而，设计客观考试题来评估学生的创造力、解决问题的能力和领域知识，仍是一项重大挑战。本文提出了一种利用 ChatGPT 提示工程技术应对这些挑战的方法。提示工程是一种高效技术，可优化语言模型的性能，无需额外微调模型，即可自动生成类型多样、难度等级各异的高质量考试题目。本研究运用多种模式和模板生成兼具理论与实践部分的考试题目，从而全面评估学生对理论的理解以及实际编程熟练度。研究通过问卷调查验证了所提方法；尽管某些方面仍有改进空间，但整体结果证实了该方法的重要性和适用性。 生成的问题和参考答案质量可与人工编写的题目相媲美，甚至更胜一筹，同时显著减少了准备题目所需的时间和精力。本研究表明，通过提示工程自动生成考试题目，能够提升教育评估工具的质量和效率，使其成为未来教育环境中的宝贵资源。

索引词—提示工程、生成式人工智能、考试题目与答案、编程课程

I. 引言

P提示工程已成为有效利用人工智能（AI）语言模型的一项关键技术 [1]-[3]。它指的是.....的过程

设计和优化提供给人工智能系统的指令或查询，即提示词，以实现预期结果。随着 ChatGPT 等大语言模型（LLM）的发展， [4]， 提示工程的重要性日益凸显。提示词的质量直接影响模型输出的准确性和相关性[5]。LLM 是经过训练的神经网络架构，能够理解人类意图和上下文信息[6]。措辞或结构上的细微差异都可能导致截然不同的输出 [7],，这凸显了精确、审慎地设计提示词的必要性。

提示工程的一项主要优势是，无需额外微调即可修改模型的输出格式和准确性 [8]。这既降低了计算成本和时间，也使用户能够更直接地控制模型行为 [9]，[10]。此外，信息丰富的提示词能够增强推理能力并提高输出质量，从而弥补当前模型固有的局限性。这在法律咨询或技术文档等对可靠性和准确性要求极高的领域尤其有价值。然而，大型语言模型的内部机制仍不透明，因此难以定量评估提示词设计与模型性能之间的关系。令牌长度限制也会制约提示词的复杂性 [11]。因此，有效的提示工程需要反复试验和定性评估。

随着人工智能日益融入日常生活和专业实践，提示工程的作用也在不断扩大 [12]-[15]。掌握这项技能有助于提升个人和组织层面的生产力，并已成为人工智能驱动的行业中的一项核心能力 [16]。这表明，有必要在教育和专业情境中系统地探索提示工程。

成功应用提示工程离不开若干基本原则 [17]-[19]。提示词应包含清晰、详细的指令，同时避免含糊或过于笼统的表达。提示词应明确界定任务目标，并在需要时提供补充上下文或参考资料，以提高准确性和相关性。此外，指定结构化输出格式和风格有助于确保内容的一致性和高质量 [20]。

ChatGPT [4] 在理解上下文和预测后续句子方面表现出色，但其应用受到令牌限制的约束。另一个局限是幻觉问题 [21]，，即模型生成看似真实、实则不准确或捏造的内容。当用户缺乏足够的领域专业知识来验证输出时，这个问题尤其突出。不过，在编程情境[22], 中，幻觉的风险较低，因为生成的代码可以执行和验证。因此，ChatGPT[4] 适用于多个领域 [23]，，并且

Jongwook Si 任职于韩国庆尚北道龟尾市 39177 的计算机人工智能融合工程系（电子邮箱：jwsi425@kumoh.ac.kr）。Sungyoung kim 任职于韩国庆尚北道龟尾市 39177 的计算机工程系（电子邮箱：sykim@kumoh.ac.kr）

在教育[24]领域尤其具有巨大潜力。

近年来，计算机编程教育不仅成为中学和高中教育的重要组成部分，也日益成为高等教育的重要内容。传统上，C 语言和 C++ 一直是基础编程语言，但 Python 凭借简单易学、语法直观等特点迅速普及。然而，语言模型的广泛应用导致学生依赖人工智能生成的代码，却没有掌握核心编程概念的基础知识，从而阻碍了创造性思维和问题解决能力的发展。大学考试主要旨在评估学生的理论知识和创造力，但手动编制高质量考试题目耗时较长，且高度依赖教师的专业知识 [25]。此外，要求学生从头编写完整程序会带来评分方面的挑战，也无法充分反映他们对理论知识的理解。因此，越来越需要通过多种题型评估学习者的综合能力。

编程教育必须培养创造性思维和问题解决能力，因此需要一种兼顾理论知识与实践经验的评估方法。然而，传统教育系统对这种方法的支持仍不充分。本研究提出一种方法，利用 ChatGPT [4] 和提示工程技术，自动生成高质量、富有创意且形式多样的编程考试题。为此，研究设计了涵盖理论与实践问题的模板，以生成难度和形式各异的题目。通过这种方法，可以全面评估学习者的知识和实践技能。该方法旨在突破现有教育评估的局限，并探索人工智能驱动的自动生成考试题目的实践潜力。

本研究的方法流程包括三个主要阶段。首先，运用提示工程技术设计模板，以生成多种类型的编程考试题。在这一过程中，纳入了题型分类、题目与答案生成规则，以及基于教育理论（建构主义和布卢姆分类法）的设计原则。其次，应用所设计的模板，自动生成涵盖多种编程语言的不同问题，包括 Python、C语言++和 Java，并对生成的题目进行系统收集和整理。第三，对主修编程的本科生开展调查，以实证评估所提出的问题生成方法的实用性和教育效果。参与者按照编程经验水平（初学者、中级、高级）进行分类，并将此作为准对照组，以便比较回答并识别各组特有的差异。涉及参与者的实证评估在研究第三阶段开展，学生直接回答自动生成的问题，或评价题目的感知质量和清晰度。通过这一过程收集的数据采用描述性统计和亚组比较进行分析。

第 2 节概述了可为编程课程设计的题目类型。第 3 节介绍了通过应用提示工程技术开发的模板，并详细说明其结构和功能。第 4 节分析了生成结果，评估其质量及其与教育目标的相关性。最后，第 5 节讨论了未来研究方向，并指出所提方法的潜在改进与拓展。

II. 编程课程背景

A. 编程教育中的选择题

在编程教育中，选择题已被证明是评估学生基础理论知识和编程技能的有效方法 [26]。选择题要求学生在有限时间内做出明确判断，使教育工作者能够高效地评估大批学生。此外，教师可以通过评估学生对特定概念、算法思维和问题解决能力的理解，全面诊断其学习状况。从高阶思维框架的角度来看，设计得当的选择题可以促进分析、评价和应用，而非仅仅考查记忆 [27]。因此，构建得当的选择题能够鼓励学生追踪代码执行过程并考虑不同情境，从而促进更高层次的认知参与 [28]。

编程教育中的选择题可分为十种类型，每种类型都针对特定的认知与实践技能。第一种要求选择正确的输出结果或找出代码中的错误，以培养代码跟踪和调试能力。第二种要求针对代码行为选出多个正确答案，鼓励学生考虑不同的执行路径。第三种要求按顺序排列代码或算法步骤，以强化逻辑与过程性思维。第四种要求补全缺失的代码元素，以帮助回忆语法并加深对功能的理解。第五种侧重于为问题选择最佳算法，以培养实际应用中的问题解决能力。第六种要求将编程术语与其定义配对，确保概念清晰。第七种要求从代码片段中组合出完整程序，以增进对模块化和程序结构的理解。第八种要求解释程序的输出结果，以培养分析能力。第九种要求选择与代码行为相符的描述，以评估理论理解。第十种评估根据代码修改预测程序行为变化的能力。这些题目类型综合考查广泛的知识，从语法和逻辑到高阶思维及实际应用。

B. 简答题

简答题分为六类。第一种要求学生跟踪给定代码的执行过程并预测其输出，以检验他们对循环、条件语句、函数调用和递归等核心概念的理解。第二种要求解释程序的用途和行为，并补全缺失代码，从而同时强化代码分析和实现技能。第三种提供表格或图表等数据可视化内容，并要求学生编写代码处理数据，从而将数据解读与编程实践结合起来。第四种只提供输出结果，并要求学生编写能够生成该结果的代码，以培养创造力和逻辑问题解决能力。第五种要求编写满足特定条件的程序，训练学生系统地分析需求并有效实现。第六种要求编写一个集成多项函数和要求的完整程序，全面评估学生在程序设计、实现和函数集成方面的高级技能。这些简答题使学习者不仅能够展示编程语法知识，还能展现逻辑推理能力和系统性问题解决能力。

C. 论述题 论述题

分为三类。第一类要求定义特定的编程术语，并解释其含义和应用，以评估学习者的概念理解能力以及将理论联系实际的能力。第二类涉及分析代码错误、描述其原因并提出解决方案，以评估学习者的调试技能和逻辑推理能力。第三类要求学生分析给定代码中的低效之处或缺陷，并提出改进方案，从而培养他们在优化和重构方面的技能，同时鼓励他们考虑性能、可读性和可维护性。

综合来看，选择题、简答题和论述题反映了从知识获取到分析、评价和创造性问题解决等不同层次的学习目标。因此，利用提示工程生成此类多样化题目，不仅限于实现评估自动化，还能以均衡且系统的方式，作为一种教育机制，提升学习者的高阶思维技能和实际编程能力。

III. 基于提示工程的拟议模板

提示工程是为编程课程创建可靠且有效的考试题目的有力方法。借助 ChatGPT 等大语言模型的能力，教育工作者可以设计与教育目标相符的高质量题目。本节探讨能够改进考试题目生成过程的关键提示工程技术，并详细说明其潜在优势与局限。图1展示了所提出方法的整体流程。

A. 角色设定

角色设定是指在提示词中指定特定角色或职业，引导语言模型生成体现该角色专业知识的回答。例如，设定“编程语言专家”“计算机科学教授”或“软件工程师”等角色，可以让模型生成更专业且符合语境的问题。这项技术旨在为模型提供上下文线索，帮助它更好地理解任务或问题。

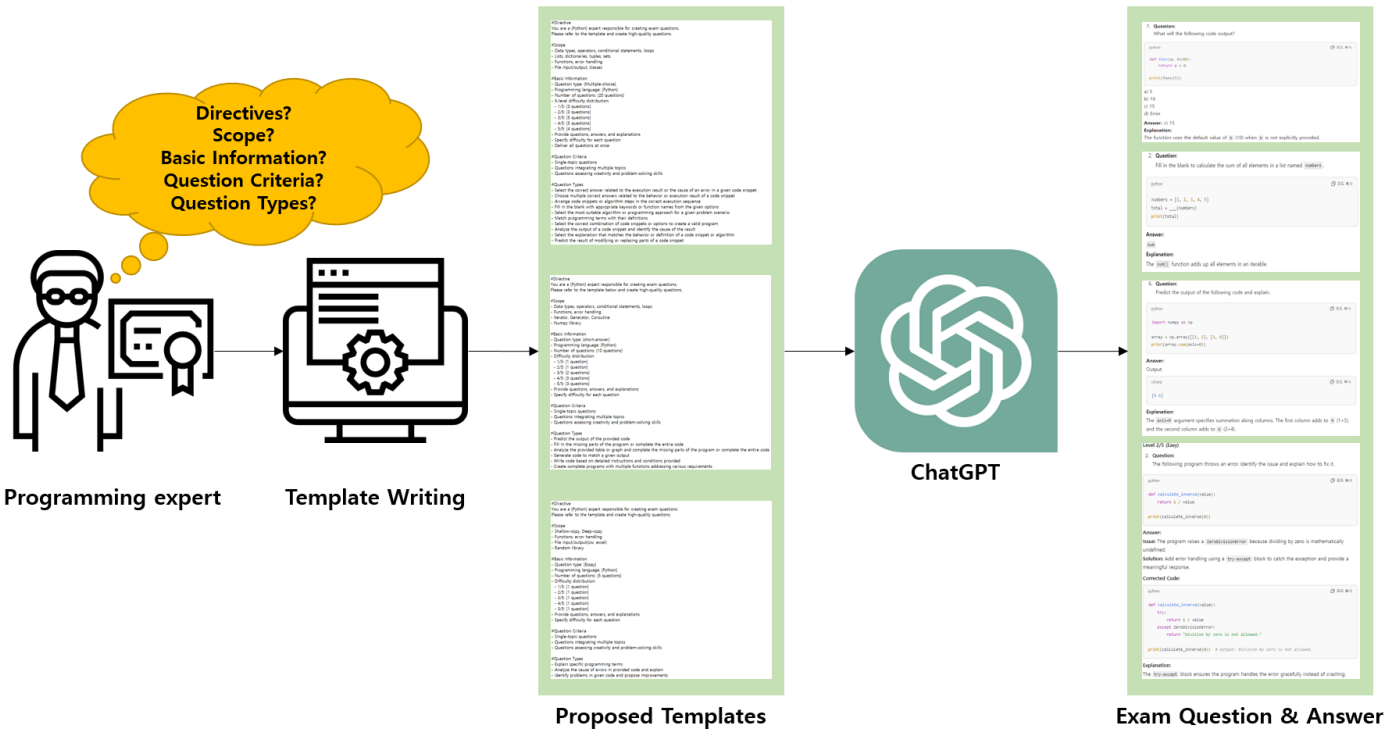


图 1. 所提方法的总体流程图。

通过设定角色，模型更有可能参考相关知识并生成高级问题和答案。例如，如果设定“编程专家”这一角色，并在提示词中要求“创建有关 Python 列表推导式的选择题”，生成的问题很可能体现出对该主题的深入理解，并符合学术语境。

然而，这项技术的一个显著局限是存在幻觉风险，即模型可能生成不准确或与指定角色不一致的信息。因此，必须仔细验证和审查生成的问题，以确保其质量和准确性。尽管存在这一局限，只要进行适当监控，角色设定仍然是一种很有吸引力的技术，可用于生成高质量、特定领域的问题。

B. 少样本学习

少样本学习是指在提示中提供少量示例，以增强模型生成更相关、上下文更丰富内容的能力。与不提供任何示例的零样本学习不同，少样本学习会使用两个或更多示例，帮助模型推断所需输出的结构或上下文。

在编程考试中，这项技术在创建多样且高质量的问题方面尤其有效。通过加入题目类型示例，例如选择题或填空题，模型能够更好地理解所需输出的格式和要求。例如，教育工作者可以在提示中提供一道选择题样例和一道简答题样例，帮助模型生成更多既多样又相关的问题。

融入少样本学习，还可以纳入特定细节，例如第 2 节中讨论的问题类型。例如，提供一段说明问题类型、难度等级和主题范围的背景信息，可确保模型生成的问题符合教育目标。与零样本提示词相比，这种技术能够生成更细致、更全面的问题。

通过提供多个示例，少样本学习显著提升了生成的问题的质量和多样性，对于旨在有效评估学生批判性思维和问题解决能力的教育工作者而言，这是一项不可或缺的工具。

C. 深津式提示

深津式提示是一种结构化方法，通过设计格式明确且带有模板的提示词，确保输出一致且高质量 [30]。这种技术将指令、范围、基本信息、题目标准和题目类型等多个组成部分整合到一个连贯的提示模板中。

本研究提出的模板由以下部分组成：

- #指令：本节指定角色和任务。例如，可能包含“编程专家”“编程教授”等角色，以及“为 Python 考试生成高级算法题”等任务。
- #范围：本部分用于界定考试题目的范围，例如 Python 数据结构、算法或面向对象编程。
- #基本信息：包括题目形式（选择题、简答题或论述题）、编程语言、题目数量和难度等级（例如，1/5~ 5/5）等详情。
- #题目标准：本节说明要生成的题目的内容和性质，例如题目是聚焦于单主题问题、多部分问题，还是创造力和问题解决能力。
- #题目类型：指定要包含的题目类型，例如预测代码输出、填补缺失代码、分析错误或结合多个概念。

使用此模板，教育工作者可以以结构化的方式提供所有必要详情，确保模型生成可靠且高质量的考试题目。例如，#指令下的角色设定有助于确立模型的视角，而加入范围和题目标准则能确保生成的问题符合预期的学习目标。此外，指定第 2 节中的题目类型，可以更好地控制问题的多样性和形式。

这种结构化方法确保模型生成的问题不仅质量高，而且符合课程的具体要求。它还能简化工作流程，使教育工作者能够高效地生成各种考试题目及其详细解决方案。

图 2 展示了所提出的选择题提示模板，图 3 展示了简答题模板，图 4 则展示了论述题模板；其中，范围和基本信息作为示例纳入其中，以便进行演示。题目类型包括第 2 节中提到的所有类型。

选择题模板旨在快速有效地评估学生对基本概念的理解和逻辑推理能力。题目采用清晰的选项结构，使学生能够针对给定情境选择最合适的答案。该模板的主要特点是提供明确无歧义的选项，在保持清晰度的同时涵盖从初学者到高级的各种难度等级，从而实现全面评估。题目类型包括预测代码执行结果、按正确顺序排列代码片段以及填写关键字。该模板尤其适用于以直接而有效的方式评估学生对概念的理解、代码分析能力和错误识别能力。

简答题模板旨在鼓励学生阐述问题解决过程并展现创造力。该模板不仅要求给出

简单正确的答案，还促使学生运用逻辑化和结构化方法解决题目。题目会提供明确的条件和指导，使学生能够根据给定的背景编写或分析代码。简单题可能要求预测简单的输出值，而较难的题目则可能要求实现复杂逻辑。主要题目类型包括填补代码空缺、根据分析得出的输出编写代码，以及通过分析表格或图表来补全程序。简答题模板适用于深入评估学生的问题解决能力、数据处理技能和逻辑设计能力。

```
#Directive
You are a [Python] expert responsible for creating exam questions.
Please refer to the template and create high-quality questions.

#Scope
- Data types, operators, conditional statements, loops
- Lists, dictionaries, tuples, sets
- Functions, error handling
- File input/output, classes

#Basic Information
- Question type: [Multiple-choice]
- Programming language: [Python]
- Number of questions: [20 questions]
- 5-level difficulty distribution:
  - 1/5: [3 questions]
  - 2/5: [3 questions]
  - 3/5: [5 questions]
  - 4/5: [5 questions]
  - 5/5: [4 questions]
- Provide questions, answers, and explanations
- Specify difficulty for each question
- Deliver all questions at once

#Question Criteria
- Single-topic questions
- Questions integrating multiple topics
- Questions assessing creativity and problem-solving skills

#Question Types
- Select the correct answer related to the execution result or the cause of an error in a given code snippet
- Choose multiple correct answers related to the behavior or execution result of a code snippet
- Arrange code snippets or algorithm steps in the correct execution sequence
- Fill in the blank with appropriate keywords or function names from the given options
- Select the most suitable algorithm or programming approach for a given problem scenario
- Match programming terms with their definitions
- Select the correct combination of code snippets or options to create a valid program
- Analyze the output of a code snippet and identify the cause of the result
- Select the explanation that matches the behavior or definition of a code snippet or algorithm
- Predict the result of modifying or replacing parts of a code snippet
```

图 2. 拟议选择题模板

论述题模板旨在评估学生的概念理解和问题分析技能。该模板引导学生以逻辑方式分析问题、描述发现并提出改进建议。题目结合了理论和实践两个方面，包括定义和解释术语等理论问题，以及识别和改进代码问题等实践问题。简单题侧重于解释概念，而较难的题目则要求分析并优化复杂代码。主要题目类型包括定义并解释特定编程术语、分析并解释代码错误的原因，以及识别给定代码中的问题并提出改进建议。论述题模板有助于学生深入理解概念、有效运用知识，并在问题解决过程中增强逻辑推理能力。

A. 结果分析

图5展示了使用所提出的选择题模板生成的问题示例，体现了该模板在评估学生的基础概念理解和逻辑推理能力方面的有效性。这些问题结构清晰简洁，使学生能够轻松理解每道题的意图。例如，考查变量命名约定的问题直接评估基础编程语法，并通过定义明确的选项尽量减少歧义。生成的问题与学习目标紧密契合，每道题都针对特定的编程能力。变量命名约定问题评估对基本语法的理解，默认参数问题鼓励学生深入理解函数设计，而文件输入/输出问题则侧重于实用的数据处理技能。这些内容涵盖从基础到中级的概念，凸显了该模板适应不同学习者熟练程度的能力。

```
#Directive
You are a [Python] expert responsible for creating exam questions.
Please refer to the template below and create high-quality questions.

#Scope
- Data types, operators, conditional statements, loops
- Functions, error handling
- Iterator, Generator, Coroutine
- Numpy library

#Basic Information
- Question type: [short-answer]
- Programming language: [Python]
- Number of questions: [10 questions]
- Difficulty distribution:
  - 1/5: [1 question]
  - 2/5: [1 question]
  - 3/5: [2 questions]
  - 4/5: [3 questions]
  - 5/5: [3 questions]
- Provide questions, answers, and explanations
- Specify difficulty for each question

#Question Criteria
- Single-topic questions
- Questions integrating multiple topics
- Questions assessing creativity and problem-solving skills

#Question Types
- Predict the output of the provided code
- Fill in the missing parts of the program or complete the entire code
- Analyze the provided table or graph and complete the missing parts of the program or complete the entire code
- Generate code to match a given output
- Write code based on detailed instructions and conditions provided
- Create complete programs with multiple functions addressing various requirements
```

图 3. 拟议简答题模板

```
#Directive
You are a [Python] expert responsible for creating exam questions.
Please refer to the template and create high-quality questions.

#Scope
- Shallow-copy, Deep-copy
- Functions, error handling
- File input/output(csv, excel)
- Random library

#Basic Information
- Question type: [Essay]
- Programming language: [Python]
- Number of questions: [5 questions]
- Difficulty distribution:
  - 1/5: [1 question]
  - 2/5: [1 question]
  - 3/5: [1 question]
  - 4/5: [1 question]
  - 5/5: [1 question]
- Provide questions, answers, and explanations
- Specify difficulty for each question

#Question Criteria
- Single-topic questions
- Questions integrating multiple topics
- Questions assessing creativity and problem-solving skills

#Question Types
- Explain specific programming terms
- Analyze the cause of errors in provided code and explain
- Identify problems in given code and propose improvements
```

图 4. 拟议描述题模板

IV. 实验

此外，为每个问题提供详细解释，能显著促进学习过程。这些解释阐明了各个选项为何有效，并帮助学生理解错误答案的问题所在。这

些反馈使学习者得以巩固概念知识，并弥补理解上的不足。例如，关于默认参数的问题清晰地解释了参数行为，确保学习者牢固掌握这一概念。最后，生成的问题在难度和主题覆盖范围方面都体现出适当的多样性，既涵盖基础语法和结构问题，也包括函数设计和文件处理等更具实践性的应用，展现了该模板对初学者和高级学习者的实用价值。



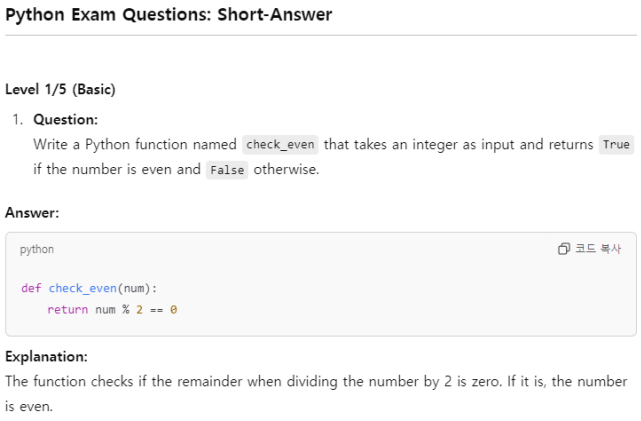
18. **Question:**
What does the following code do?

```python
with open("data.txt", "w") as file:
    file.write("Hello, World!")
```

a) Reads the content of `data.txt`
b) Appends "Hello, World!" to `data.txt`
c) Overwrites `data.txt` with "Hello, World!"
d) None of the above

Answer: c) Overwrites `data.txt` with "Hello, World!"
Explanation:
The `w` mode overwrites the file.

图5. 生成的选择题及答案示例



Python Exam Questions: Short-Answer

Level 1/5 (Basic)

1. **Question:**
Write a Python function named `check_even` that takes an integer as input and returns `True` if the number is even and `False` otherwise.

Answer:

```python
def check_even(num):
    return num % 2 == 0
```

Explanation:
The function checks if the remainder when dividing the number by 2 is zero. If it is, the number is even.

图6. 生成的简答题及答案示例

总之，选择题模板生成的问题在清晰度、与学习目标的一致性以及反馈提供方面都非常有效。该模板能够适应不同的学习水平，证明是教育评估的有力工具。

如图 6 所示，使用所提出的简答题模板生成的问题旨在评估学生的问题解决能力、逻辑推理能力和编程熟练度。每道题都经过清晰编排，确保学生能够轻松理解并完成题目所提出的任务。例如，要求编写一个函数来判断数字是否为偶数的问题，将学生的注意力集中在一项具体任务上；而计算列表总和的填空题，则能有效评估学生对 Python 内置函数的理解。

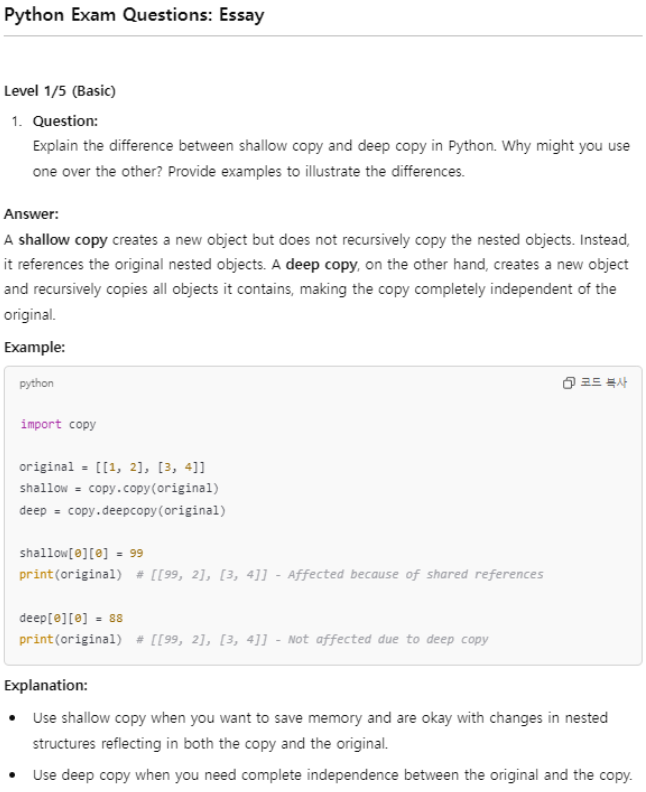
生成的问题涵盖了广泛的主题，从基础编程概念到更高级的内容。基础题目（例如编写函数检查偶数或计算列表总和）有助于学生巩固对 Python 语法和函数用法的理解。相比之下，涉及 NumPy 库的题目（例如计算数组总和并解释其行为）则能评估学生在中级水平上的实际数据处理和数组操作能力。这种多样性表明，该模板适用于不同熟练程度的学习者，从初学者到中级学生。此外，简答题模板鼓励学生分析题目并编写恰当的代码来解决问题。例如，填空题会考查学生有效识别并应用 Python 特定功能的能力。NumPy 题目要求学生理解并解释 axis 参数的行为，为学生提供了亲自动手实践真实世界数据操作技术的机会。

每道题都附有详细的答案和解释，显著促进了学习过程。例如，偶数判断函数问题解释了如何使用取模运算符来判断是否能被二整除。类似地，NumPy 数组问题详细说明了 axis 参数如何用于计算各列的和，帮助学生理解多维数组的行为。这些解释非常有助于巩固概念，并帮助学生从错误中学习。生成的问题还涵盖了多种情境，例如函数实现、补全代码片段以及预测代码的输出。这些情境反映了现实世界中的编程挑战，并弥合了理论学习与实际应用之间的差距。引入 NumPy 等库可确保学生接触到与行业及实际情境相关的工具和技术。

总之，简答题模板是一种强大的工具，旨在帮助学生运用编程技能并展示问题解决能力。题目清晰、具有教育意义，并兼顾基础与高级主题。此外，每道题都附有详细解析，使该模板成为评估与学习的宝贵资源，也证明了它作为评估和提升编程教育水平的工具的有效性。

使用所提出的论述题模板生成的问题旨在深入评估学生的概念理解和问题分析技能，如图 7 所示。这些问题并非仅仅要求给出正确答案，而是促使学生分析问题的根本原因、提出解决方案，或清晰地解释概念。例如，第一道问题要求学生解释浅拷贝和深拷贝的区别，以及各自的使用场景。这有助于学生加深对复杂数据结构和内存管理的理解。第二道问题给出了一段代码片段，

该代码片段会触发 ZeroDivisionError，并要求学生提出恰当的异常处理策略。此类问题用于评估学生分析错误并有效运用 Python 异常处理机制的能力。



Python Exam Questions: Essay

Level 1/5 (Basic)

1. **Question:**
   Explain the difference between shallow copy and deep copy in Python. Why might you use one over the other? Provide examples to illustrate the differences.

Answer:

A **shallow copy** creates a new object but does not recursively copy the nested objects. Instead, it references the original nested objects. A **deep copy**, on the other hand, creates a new object and recursively copies all objects it contains, making the copy completely independent of the original.

Example:

```python
import copy

original = [[1, 2], [3, 4]]
shallow = copy.copy(original)
deep = copy.deepcopy(original)

shallow[0][0] = 99
print(original)  # [[99, 2], [3, 4]] - Affected because of shared references

deep[0][0] = 88
print(original)  # [[99, 2], [3, 4]] - Not affected due to deep copy
```

Explanation:

- Use shallow copy when you want to save memory and are okay with changes in nested structures reflecting in both the copy and the original.
- Use deep copy when you need complete independence between the original and the copy.

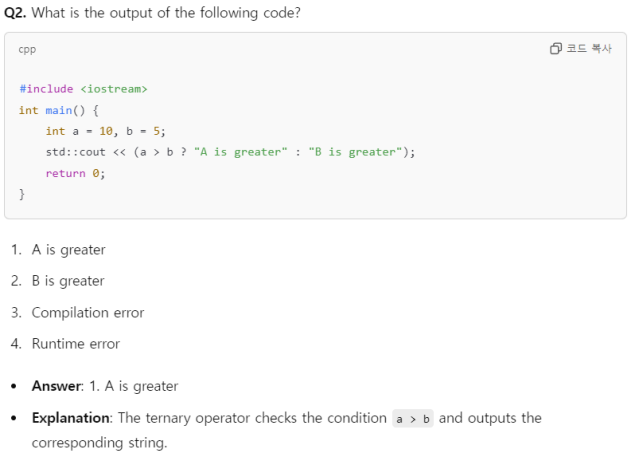
图 7. 生成的论述题及答案示例

生成的问题经过精心设计，旨在引导学生有条理地解决问题并给出详细解释。第一题要求学生比较浅拷贝和深拷贝的行为，并使用示例代码加以说明，从而全面理解数据结构复制在实践中的工作方式。第二题要求学生找出错误，并使用 try-except结构提出解决方案，帮助他们培养实用的调试和错误处理技能。

每道问题都附有详细的答案和解释，帮助教师评分并比较作答情况。关于浅拷贝和深拷贝的问题，其解释清楚地阐明了每个概念及其行为，并通过示例代码和结果加深理解。同样，关于异常处理的问题，其解释概述了错误产生的原因，并提供了修正后的代码，同时说明 try-except结构如何提升程序的稳定性。如此详细的反馈对于帮助学生从错误中学习并巩固对关键概念的理解非常有效。论述题还涵盖广泛的主题，并提升实际问题解决能力。浅拷贝和深拷贝的问题要求全面理解内存管理和数据结构的行为；在处理复杂数据结构时，这些都是至关重要的技能。在实践中，异常处理问题评估学生确保代码健壮性的能力，涉及现实编程中常见的问题。

总之，论述题模板是一种有效的工具，可用于评估学生分析问题、有条理地提出解决方案并运用实用编程技能的能力。问题表述清晰，并与教育目标紧密契合，同时兼顾基础内容与高级主题，以满足多样化的学习目标。此外，详细的解释提供了有价值的学习支持，表明论述题模板既可有效用作评估工具，也可用作教学工具。

前文讨论聚焦于 Python，而这里将模板应用于其他编程语言，具体包括 C++ 和 Java，并对结果进行比较。Java 的结果如图 8 所示，C++的结果如图 9 所示。将所提出的多项选择题模板应用于 C++ 和 Java，可以看出每种语言独特的语法和教育目标如何影响生成的问题的侧重点和结构。这表明，该模板并不局限于特定语言，可以灵活地适用于各种编程语言。



Q2. What is the output of the following code?

```cpp
#include <iostream>
int main() {
    int a = 10, b = 5;
    std::cout << (a > b ? "A is greater" : "B is greater");
    return 0;
}
```

1. A is greater
2. B is greater
3. Compilation error
4. Runtime error

- **Answer**: 1. A is greater
- **Explanation**: The ternary operator checks the condition `a > b` and outputs the corresponding string.

图 8. 转换为 C语言的生成结果示例++



Q5. What will happen if the following code is executed?

```java
public class Main {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3};
        System.out.println(numbers[3]);
    }
}
```

1. Prints `0`
2. Prints `null`
3. Throws an `ArrayIndexOutOfBoundsException`
4. Throws a `NullPointerException`

- **Answer**: 3. Throws an `ArrayIndexOutOfBoundsException`
- **Explanation**: The array index `3` is out of bounds since the valid indices are `0, 1, 2`.

图 9. 转换为 Java 的生成结果示例

C++ 问题着重体现该语言的结构特点和内存管理等底层编程概念。例如，一道要求使用三元运算符判断两个值中较大值的问题，可以评估学习者对运算符优先级和条件替换的理解。这类问题有助于进行精准的代码分析并理解执行流程，帮助学习者掌握高性能系统编程所需的技术技能。

Java问题侧重于面向对象编程（OOP）的原理和运行时错误处理。例如，关于访问越界数组索引导致数组索引越界异常的问题，能够有效评估Java严格的运行时规则和异常处理机制。这些问题强调结构完整性和代码稳定性，使学习者掌握解决现实软件开发中常见运行时问题所需的技能。

生成的问题体现了每种语言的独特特性和学习目标，展现了模板的通用性。C++ 问题要求熟练掌握内存管理和高性能编程，面向中级和高级学习者。Java问题则涉及运行时错误和面向对象设计原则，评估学习者编写结构清晰、可靠代码的能力。

B. 调查评估 本研究进行了

一项针对编程专业本科生的调查，以评估所提出的问题模板的实用性和教育成效。调查包含14个问题，分为四类：模板功能（#1-#5）、问题与答案评估（#6-#10）、问题自然度（#11-#12）以及学习成效和应用意向（#13-#14）。受访者使用五点李克特量表进行评分，评分范围从1（非常不同意）到5（非常同意）。调查在线开展，历时20天，共有46人参与。根据编程经验，受访者分为四组：不足1年、1-2年、2-3年以及超过3年。图10展示了根据编程经验对参与者进行分类的情况。表1列出了拟定的调查问题，表2则按编程经验分类并汇总了总体结果的均值（M）和标准差（SD）。

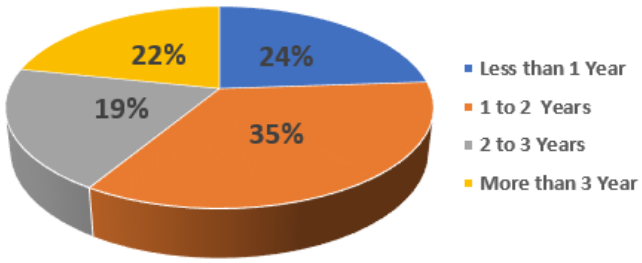


图 10. 调查参与者的编程经验

表 I. 所提方法的调查问题

	问题
#1	能够设置题目类型（例如选择题、简答题、描述题）很有用。
#2	能够指定编程语言（例如 Python、使用 C语言++、Java 进行题目生成很有帮助。
#3	区分不同难度等级（例如从 1/5 到 5/5），以便量身定制符合学习水平的题目是有效的。
#4	一次设置题目数量（例如 10 或 20 道问题）的功能很有优势。
#5	生成各种类型题目（例如，错误检测、预测代码输出、代码补全）根据模板说明进行操作是切实可行的。
#6	生成的题目具有足够的多样性。
#7	问题陈述的解释清晰且易于理解。
#8	题目中融入的编程知识恰当且经过充分推敲。
#9	答案（或解决方案）逻辑上成立，且没有错误。
#10	这些题目非常适合用于评估所学材料。
#11	生成的题目在质量上可与以下题目相媲美由人类创建的题目。
#12	题目中的措辞和表达自然且不显得生硬。
#13	使用模板应对不同难度的题目和题型，有助于提高相关课程的成绩。
#14	将生成的内容用于今后的考试（例如期中考试或期末考试）被认为是一种积极的做法。

总体而言，所有组别对人工智能题目生成系统的评价均为正面；但在具体项目上存在明显差异。值得特别注意的是，学习时间不足一年的组别始终呈现较大的标准差，表明其回答缺乏一致性。这说明初学者缺乏经验，导致他们的认知更加多变且不稳定。

#1（设置题型的能力）的总体平均分为 4.30（0.78），表明评价较为正面。从各组结果来看，学习时间为 2 至 3 年的组别评分最高，为 4.67（0.50），认为选择题型对其学习过程有实质帮助。学习时间为 1 至 2 年的组别也给予正面评价，评分为 4.44（0.63）。相比之下，学习时间不足一年的组别得分最低，为 3.91（1.14），且标准差最大，说明学习经验有限的初学者难以有策略地利用题型之间的差异，意见也存在较大分歧。这可以用建构主义学习理论来解释：缺乏先前经验时，学习者无法充分理解新工具的意义。

2（指定编程语言的能力）获得了最高的总体评价，平均分为 4.53（0.62）。2 至 3 年组给出的评价最高，为 4.89（0.33），这反映出一种看法：提供针对特定语言的问题，能够直接助力实现学习目标。1 至 2 年组的评价也很高，为 4.60（0.51）；3 年及以上组紧随其后，评分稳定在 4.40（0.52）。尽管

不足 1 年组的评分较低，为 4.27（0.79），但仍持正面态度。这些结果与自我导向学习理论一致，该理论强调，当学习者选择符合自身目标的学习资源时，会更有动力，也更满意。

#3（调整难度等级的能力）的总体平均分为4.07（0.81），相对略低。学习年限不足1年的组和2—3年的组均评分为4.00（0.89、1.00），而学习3年或以上的组评分略高，为4.10（0.57）。学习1—2年的组相对更为积极，评分为4.13（0.72）。这表明，初学者可能会觉得难度区分较为模糊，而高级学习者更加重视题目的深度和真实性，而非预设等级。从认知负荷理论的角度来看，初学者可能会将不必要的难度区分视为外在负荷。

#4（设置问题数量的能力）获得了较高评价，总体平均分为4.43（0.65）。2—3年的组评分最高，为4.56（0.53），其次是1—2年的组，为4.50（0.63）。学习3年或以上的组评分为4.30（0.48），学习年限不足1年的组为4.36（0.81）。所有组均对该项目给予了积极评价。这一发现可以与自我调节学习理论联系起来，因为控制工作量的能力是促进学习者自主性的重要工具，尤其是在中级阶段。

#5（生成多样化问题类型的能力）的总体平均分为 4.44（0.69）。1-2 年组的评分最高，为 4.69（0.48）；3 年及以上组的评分也很高，为 4.67（0.50）。2-3 年组的评分为 4.56（0.53）。相比之下，不足 1 年组的评分最低，为 3.82（0.98），且标准差最大。这表明，初学者往往将多样性视为复杂性，而中级和高级学习者则重视多样性，认为它有助于维持学习动机并促进知识迁移。

#6（题目多样性）的总体平均分为 4.09（0.72）。3 年及以上组的评分最高，为 4.30（0.48）；其次是 2-3 年组，为 4.22（0.67）；1-2 年组的评分为 4.06（0.70）。不足 1 年组的评分最低，为 3.82（0.87）。这表明，初学者更重视可理解性而非多样性，这与认知负荷理论相符：学习者力求减少外在负荷。

#7（问题陈述的清晰度）的总体平均得分为 4.07（0.74）。2–3 年组的评分最高，为 4.33（0.50）；其次是 3 年及以上组，为 4.10（0.57），1–2 年组为 4.00（0.74），不足一年组为 3.91（0.83）。这表明，已经认为题目难度较高的初学者对表述不清尤为敏感。

#8（编程知识的适切性）的总体平均分为 4.28（0.66）。3 年及以上组的评分最高，为 4.50（0.53）；其次是 2-3 年组，为 4.33（0.50），以及 1-2 年组，为 4.31（0.70）。不足 1 年组的评分最低，为 4.00（0.77）。这表明，高级学习者更关注题目是否符合实际编程知识和课程体系。

#9（答案／解决方案的逻辑有效性）的总体平均分为 4.09 （ 0.75 ）。有趣的是，不足 1 年组的评分相对较高，为 4.27（0.90），这可能是因为他们评判逻辑错误的能力有限。相比之下，1-2 年组的评分最低，为 3.81（0.81），这反映出该阶段的学习者更加关注推理的准确性和一致性。

#10（适用于评估）获得了较为正面的评价，总平均分为4.20（0.59）。3年及以上组的评分最高，为4.40（0.52），其次是2至3年组，为4.22（0.67）。不足1年组的评分最低，为3.91（0.83），这表明高级学习者比初学者更能认识到题目的评估效用。

#11（与人工创建的问题进行比较）的总体评分最低，平均分为3.89（0.61）。值得注意的是，不足1年组的评分最低，为3.55（0.69）。这可能不只是因为经验不足，也可能是因为初学者缺乏比较人工智能生成的题目与人工创建的问题所需的评价框架，因而觉得前者陌生或不完整。1至2年组的评价较为正面，评分为4.19（0.64）；而2至3年组和3年及以上组的评分则再次降低，分别为3.89（0.60）和3.80（0.42）。这表明中级学习者往往采取更务实的看法，而高级学习者的评价则更为严谨，着重考量创造力、情境适切性和完整性。

#12（措辞和表达的自然度）评分较高，总体平均分为 4.35（0.66）。3 年及以上组的评分最高，为 4.50（0.53）；即使是学习时间不足 1 年的组，也给出了 4.27（0.79）的积极评价。自然直观的表达让初学者更容易理解题目，也让高级学习者感到更加专业。

表 II. RES所提方法调查结果（按编程经验划分的详细分数及总分 s)

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
1 年↓	3.91 (1.14)	4.27 (0.79)	4.00 (0.89)	4.36 (0.81)	3.82 (0.98)	3.82 (0.87)	3.91 (0.83)	4.00 (0.77)	4.27 (0.90)	3.91 (0.83)	3.55 (0.69)	4.27 (0.79)	4.09 (0.83)	3.73 (0.65)
1至2年	4.44 (0.63)	4.60 (0.51)	4.13 (0.72)	4.50 (0.63)	4.69 (0.48)	4.06 (0.70)	4.00 (0.74)	4.31 (0.70)	3.81 (0.81)	4.38 (0.63)	4.19 (0.64)	4.31 (0.60)	4.44 (0.63)	4.31 (0.70)
2 至 3 年	4.67 (0.50)	4.89 (0.33)	4.00 (1.00)	4.56 (0.53)	4.56 (0.53)	4.22 (0.67)	4.33 (0.50)	4.33 (0.50)	4.22 (0.67)	4.00 (0.71)	3.89 (0.60)	4.33 (0.50)	4.67 (0.50)	4.11 (0.60)
3 年↑	4.20 (0.63)	4.40 (0.52)	4.10 (0.57)	4.30 (0.48)	4.67 (0.50)	4.30 (0.48)	4.10 (0.57)	4.50 (0.53)	4.20 (0.42)	4.40 (0.52)	3.80 (0.42)	4.50 (0.53)	4.20 (0.42)	4.20 (0.42)
All	4.30 (0.78)	4.53 (0.62)	4.07 (0.81)	4.43 (0.65)	4.44 (0.69)	4.09 (0.72)	4.07 (0.74)	4.28 (0.66)	4.09 (0.75)	4.20 (0.59)	3.89 (0.61)	4.35 (0.66)	4.35 (0.62)	4.11 (0.64)

#13（提升学习成果的潜力）总体平均分较高，为4.35（0.62）。2—3 年组的评分最高，为 4.67（0.50），表明中级学习者认为基于模板的问题生成带来的益处最大。相比之下，学习时间不足 1 年的组评分较低，为 4.09（0.83），说明初学者不太能充分感受到其潜在益处。

#14（考试适用性）的总体平均分为4.11（0.64）。1-2年组给出了最高评分4.31（0.70），其次是3年及以上组的4.20（0.42）和2-3年组的4.11（0.60）。不足1年组的评分最低，为3.73（0.65），这反映出他们缺乏考试经验，对评估情境的认识也较为有限。

总的来说，虽然所有项目总体上均获得了正面评价，但不足1年组始终给出较低的平均分，且标准差较大，表明其看法存在更大的差异。这说明初学者由于缺乏经验，难以持续、准确地认识系统功能的实用性。相比之下，1-2年组和2-3年组给出的评分更高且更为一致，尤其强调了编程语言规范、工作负载控制和学习成效。这些特征与自主学习高度契合，也反映出学习者将系统功能与个人学习目标联系起来的能力不断增强。

C. 讨论

生成式人工智能的蓬勃发展，正在从根本上重塑编程教育的范式。过去，学习者主要专注于掌握编程语言的语法并手动编写代码。如今，学习者却越来越多地与 ChatGPT 等语言模型互动，以加快问题解决过程或委托其生成代码。在这一转变中，提示工程不再只是单纯的技术工具，而是成为一项核心学习能力，使学习者能够通过与人工智能互动来设计和调控自己的学习过程。

从教育视角来看，所提出方法的贡献可从多个层面加以理解。首先，提示工程要求学习者界定问题、说明背景，并为人工智能系统制定条件。这一过程构成了一种元认知活动，学习者借此将自己的思维外化并加以组织。在此过程中，学习者不仅是在寻找答案，也在分析问题情境并明确学习目标。这些实践具体体现了建构主义学习理论，该理论强调学习者主动建构知识。其次，所提出的模板在设计时明确参考了布卢姆分类法，鼓励学习者超越知识回忆和理解，进而开展分析、评价和创造性问题解决。通过要求学习者阐述推理过程、表达解决方案并探索多种方法，自动生成的问题培养了高阶思维技能。这体现了一种教学上的转变：编程教育从死记硬背转向以问题解决为中心的学习。

第三，基于提示工程的问题设计也对教师具有启示意义。传统上，编写编程考试题需要投入大量时间和专业知识，往往降低了教学实践的效率。然而，所提出的方法使教师能够高效生成多样化的问题，并使其与特定学习目标相契合，从而增强教学设计能力。此外，生成的问题结构一致、逻辑连贯，有助于提升评估实践的信度和效度。第四，所提出的模板重新诠释了评估，不再局限于传统的“正确答案是什么”这一问题，而是强调学习者如何理解问题、构建逻辑推理并清晰地呈现解决方案。这一转变有助于学习者将编程知识应用于真实情境，并阐述自己的认知过程，从而促进学习迁移。从这个意义上说，人工智能生成的考试题不仅是评估工具，也是推动深度学习的催化剂。

总之，所提出的模板在生成式人工智能驱动的编程教育背景下展现出明确的教育价值。具体而言，该模板促进学习者进行元认知反思，系统地培养高阶思维技能，支持教师提升教学设计与评估能力，并增强学习者在真实情境中应用知识的能力。因此，本研究的价值不仅在于技术创新，更在于确立了一个教学转折点，引导编程教育转向以学习者为中心、以问题解决为导向的实践。

V. 结论

本研究提出了一种运用提示工程技术的模板，用于在编程教育中有效生成考试题目及答案，并分析了该模板在各种编程语言中的应用。研究结果表明，所提出的模板是一种稳健的工具，能够通过选择题、简答题和论述题，全面评估学习者的概念理解、问题解决能力和逻辑推理。生成的问题旨在体现不同编程语言独有的语法特征和学习目标，证实该模板具有灵活性和适应性，可用于多种语言，而不局限于某一种语言。所提出的模板涵盖广泛的编程语言概念，并系统设计了题目类型，以评估学习者对这些概念的理解及其应用能力。这些问题不止要求给出简单答案，还鼓励学习者分析题目、提出恰当的解决方案，或清晰阐述概念。这种方法有助于培养学生对编程概念的深入理解和实际应用技能。

因此，自动生成的问题被

发现有助于实现实际的教育目标，而不只是提供形式上的表面多样性。在一项试点分析中，学习者表示，生成的问题帮助他们厘清概念理解，并使问题解决过程更加系统化。这表明，所提出的模板不仅是技术自动化工具，更是一种能够切实强化学习过程的机制。尽管如此，本研究仍存在一些局限。用于追踪学习成果的长期数据集仍然不足，还需要采用更严格的方法论并开展更广泛的验证，才能全面评估该模板对持续学习和实际绩效提升的影响。

未来的工作应探索以下方向。首先，开发一种能够系统地调整问题难度的自动化系统，以生成适合不同学习者熟练程度的问题，从而提高难度管理的效率。其次，改进模板设计，使其支持针对多种编程语言和学习目标优化的问题生成，这一点至关重要。采用自动化提示优化技术能够显著拓展模板的适用范围并提升其有效性。第三，应将模板从编程教育拓展至其他学科，例如数学、物理学和数据科学，在这些领域，基于提示的问题生成可作为有效的学习与评估工具。第四，开展涉及学生和教师的用户研究，对于评估模板在教育实践中的影响至关重要。此类研究可以衡量满意度和学习结果，为进一步完善和改进模板提供参考。

总之，所提出的模板作为编程教育中有效的学习与评估工具，展现出巨大潜力。它兼顾清晰度与教育一致性，有助于提升学生的问题解决能力和逻辑推理能力。未来的研究应致力于进一步增强模板的通用性和实用性，从而全面提升编程教育的质量。

参考文献

[1] M. Sidji 和 M. Stephenson，《通过提示工程让 ChatGPT 玩代号》，载于《2024年 IEEE 游戏会议（CoG）论文集》，意大利米兰，2024年8月，第1-4 页。[2] P. Bansal，《提示工程的重要性及其在生成式人工智能中的适用性》，《计算机与通信杂志》，第 12 卷，第 10 期，第 14-23 页，2024年10月，doi：10.4236/jcc.2024.1210002。[3] D. Park、G. T. An、C. Kamyod 和 C. G. Kim，《大语言模型生成式人工智能提示工程绩效提升研究》《网络工程杂志》，第 22 卷，第 8 期，第 1187-1206 页，2023年8月，doi：10.13052/jwe1540-9589.2285。[4] J. Achiam 等，《GPT-4 技术报告》，arXiv 预印本 arXiv:2303.08774，2023年3月。[5] I. L. Alberts、L. Mercolli、T. Pyka、G. Prenosil、K. Shi、A. Rominger 和 A. Afshar-Oromieh，《大语言模型（LLM）和 ChatGPT：它们将对核医学产生什么影响？》，《欧洲核医学与分子影像杂志》，第 50 卷，第 6 期，第 1549-1552 页，2023年6月，doi：10.1007/s00259- 023-06172-w。[6] X. Lin、W. Wang、Y. Li、S. Yang、F. Feng、Y. Wei 和 T. S. Chua，《面向基于大语言模型的推荐的数据高效微调》，载于《论文集第47届ACM SIGIR信息检索研究与发展国际会议，美国华盛顿特区，2024年7月，第365-374页。[7] G. Marvin、N. Hellen、D. Jjingo和J. Nakatumba-Nabende，“大语言模型中的提示工程”，载于《数据智能与认知信息国际会议论文集》，印度蒂鲁内尔维利，2023年6月，第387-402页。[8] L. Giray，“使用ChatGPT进行提示工程：学术写作者指南”，《生物医学工程年鉴》，第51卷，第12期，第2629-2633页，2023年12月，doi: 10.1007/s10439-023-03272-4。[9] B. Meskó，“提示工程作为医疗专业人员的一项重要新兴技能：教程”，《医学互联网研究杂志》，第25卷，e50638，2023年7月，doi: 10.2196/50638。[10] T. F. Heston和C. Khun，“医学教育中的提示工程”，《国际医学教育》，第2卷，第3期，第198-205页，2023年8月，doi: 10.3390/ime2030019。[11] K. Busch、A. Rochlitzer、D. Sola和H. Leopold，“直接告诉我：业务流程管理中的提示工程”，载于《业务流程建模与开发支持国际会议论文集》，西班牙萨拉戈萨，2023年5月，第3-11页。[12] U. Lee、H. Jung、Y. Jeon、Y. Sohn、W. Hwang、J. Moon和H. Kim，“少样本已足够：探索英语教育中用于自动问题生成的ChatGPT提示工程方法”，《教育信息技术》，第29卷，第9期，第11483-11515页，2024年10月，doi: 10.1007/s10639-023-12249-8。[13] C. E. Short和J. C. Short，“人工智能企业家：ChatGPT、提示工程与创业修辞创作”，《创业洞察》，第19卷，e00388，2023年6月，doi: 10.1016/j.jbvi.2023.e00388。[14] P. Korzynski、G. Mazurek、P. Krzypkowska和A. Kurasinski，“人工智能提示工程作为一种新型数字能力：对ChatGPT等生成式人工智能技术的分析”，《创业与商业经济评论》，第11卷，第3期，第25-37页，2023年11月。[15] Y. Liu等，“通过提示工程越狱ChatGPT指南”，载于《第四届软件工程、人工智能、数据质量、信息物理系统与物联网国际研讨会论文集》，巴西加利尼亚斯港，2024年7月，第12-21页。[16] L. Henrickson和A. Meroño-Peñuela，“提示意义：以诠释学方法优化ChatGPT提示工程”，《人工智能与社会》，第1-16页，2023年9月，doi: 10.1007/s00146-023-01752-8。[17] J. Park和S. Choo，“面向教育工作者的生成式人工智能提示工程：实用策略”，《特殊教育技术杂志》，2024年11月。doi: 10.1177/01626434241298954。[18] W. Cain，“提示变革：探索大语言模型人工智能中的提示工程及其变革教育的潜力”，TechTrends，第68卷，第1期，第47-57页，2024年1月。[19] N. Knoth、A. Tolzin、A. Janson和J. M. Leimeister，“人工智能素养及其对提示工程策略的影响”，《计算机与教育：人工智能》，第6卷，100225，2024年1月，doi: 10.1007/s11528-023-00896-0。[20] A. Kale等，“揭示人工智能提示工程的力量：全面探究”，载于《2024年第十届电气与能源系统国际会议》（ICEES）论文集，印度金奈，2024年8月，第1-9页。[21] H. Alkaissi和S. I. McFarlane，“ChatGPT中的人工幻觉：对科学写作的影响”，Cureus，第15卷，第2期，2023年2月，doi: 10.7759/cureus.35179。[22] N. M. S. Surameery和M. Y. Shakor，“使用ChatGPT解决编程错误”，《国际信息技术与计算机工程杂志》，第31卷，第17- 22页，2023年12月，doi: 10.55529/ijitc.31.17.22。[23] M. Haman和M. Školník，“使用ChatGPT开展文献综述”，《研究问责》，第31卷，第8期，第1244-1246页，2024年12月，doi: 10.1080/08989621.2023.2185514。[24] C. K. Lo，“ChatGPT对教育有何影响？文献快速综述”，《教育科学》，第13卷，第4期，410，2023年4月，doi: 10.3390/educsci13040410。[25] L. Rodrigues、F. D. Pereira、L. Cabral、D. Gašević、G. Ramalho和R. F. Mello，“使用GPT-4评估自动生成的简答题答案质量”，《计算机与教育：人工智能》，第7卷，100248，2024年12月，doi: 10.1016/j.caeai.2024.100248。[26] C. Grévisse，“基于GPT的选择题生成的比较质量分析”，载于《应用信息学国际会议论文集》，厄瓜多尔瓜亚基尔，2023年10月，第435-447页。[27] S. P. Stevens、S. W. Palocsay和L. J. Novoa，“分析学导论课程中选择题测试题编写实用指南”，《INFORMS教育汇刊》，第24卷，第1期，第51-69页，2023年。[28] P. P. Lemons和J. D. Lemons，“用于评估高阶认知技能的问题：不仅仅是布卢姆”，CBE—Life Sci. Educ.，第12卷，第1期，第47-58页，2013年。

[29] S. Moore、H. A. Nguyen、N. Bier、T. Domadia和J. Stamper，“使用GPT-3评估学生生成的简答题质量”，载于《欧洲技术增强学习会议论文集》，法国图卢兹，2022年9月，第243-257页，查姆：Springer International Publishing。[30] T. Fukatsu，“ChatGPT工作艺术革命”，《周刊东洋经济》，2023年4月22日，第42-43页。