

# 生成式人工智慧輔助帕森斯問題教學之探索與實例展示

## Exploring and Showcasing the Use of Generative Artificial Intelligence for Parsons Problem Instruction

徐碩亨<sup>1</sup>張智凱<sup>2</sup>

SHOU HENG, HSU<sup>1</sup> CHIH KAI, CHANG<sup>2</sup>

<sup>1</sup> 國立臺南大學 數位學習科技學系 學士生

<sup>1</sup> National University of Tainan Department of Information and Learning Technology Student

E-mail : <sup>1</sup> [asd30891750@gmail.com](mailto:asd30891750@gmail.com)

<sup>2</sup> 國立臺南大學 數位學習科技學系 教授

<sup>2</sup> National University of Tainan Department of Information and Learning Technology Professor

E-mail : [chihkai@mail.nutn.edu.tw](mailto:chihkai@mail.nutn.edu.tw)

### 摘要

在生成式人工智慧在近期快速的發展下，這項工具已儼然成為了未來的趨勢之一，也因此越來越多人開始使用這項工具在各種不同領域，包括資訊教育領域等，本研究探討運用生成式人工智慧來幫助教師生成帕森斯問題(Parsons problem)的可行性和效果，並採取設計本位研究法 (Design-Based Research, DBR)，透過與生成式人工智慧持續對話來達到預期中的效果，讓教師能夠根據學習內容製作優質的教學或學習素材，並使教師能夠專注於教學本身，提升教學效率和品質。

**關鍵字：**生成式人工智慧、帕森斯問題、資訊教育、設計本位研究法

### Abstract

With the rapid development of generative artificial intelligence, this tool has become one of the trends for the future. As a result, more and more people are using this tool in various fields, including CS education. This study explores the feasibility and effectiveness of using generative AI to help teachers generate Parsons problems and adopts a Design-Based Research approach to achieve the expected results through continuous dialogue with the generative AI. The goal is to enable teachers to create high-quality teaching or learning materials based on the content they are teaching, and to focus on teaching itself, thereby improving teaching efficiency and quality.

**Keywords:** Generative Artificial Intelligence, Parsons Problem, CS Education, Design-Based Research

# 壹、前言

隨著 ChatGPT 的誕生，生成式人工智慧導致世界各地掀起一波使用這項人工智慧技術的浪潮，它能夠自動生成新的內容或資訊，例如圖片、文字，程式碼等，用途廣泛，帶來了應用價值和創新潛力，在各個領域和行業中都能發揮重要的作用，也因這項技術能生成程式碼，現今有許多人開始運用這項技術來學習程式語言，若應用在教學層面更是可以視為精進教學的契機，AI 的作用不是要完全替代真人教師，而是要幫助他們完成一些重複性和輔助性的教學任務，從而減輕教師的負擔，解決師資短缺的問題，也協助教師更有效率的針對學習內容製作教學或學習素材，讓教學者專注於教學本身。

帕森斯問題是一種透過是一種用來輔助程式教學的一種方法，這個方法透過重新移動經過打亂的程式碼到原始的位置來訓練學生的程式語言的語法概念和邏輯思維，因此帕森斯問題的核心在於題目，在哪裡置換程式碼或是增加不同的變體都能影響學生的學習成效，這是一項複雜的作業流程，因此本研究想探討 ChatGPT 是否能輔助教師來生成帕森斯題，以及如何精確的使用提示(prompt)來使 ChatGPT 生成預期中的題目，再加以精進修改，提高教師工作效率。

## 貳、文獻探討

### 一、帕森斯問題

帕森斯問題(Parsons Problem)是一種程式設計教學方法，旨在培養學生對於程式碼邏輯思維與流程的理解能力。最初由 Dale Parsons 和 Patrica Harden 提出的概念，透過程式設計謎題(programming puzzle)來訓練學生的邏輯思維能力(Parsons & Haden, 2006)。而後經過後人多次改良，成為目前作為訓練入門計算機科學之程式學習學生的工具。而在典型的帕森斯問題中，會將一段完整的程式碼打亂順序，學生需要正確排列代碼塊並辨別符合語法與邏輯概念的程式模塊，才能使程式碼正確運行。此方法強調測試學生對於程式設計概念與邏輯思維的掌握程度，透過帕森斯問題能幫助教師發現學生在學習時所遭遇的困難，並針對學生的不理解的概念或邏輯加強練習，藉此提高學生的程式設計能力(Helminen et al., 2012)。

相較於傳統的程式教學方式，如指派撰寫程式碼(code writing)或跟蹤程式碼(code tracing)作為課後練習作業，雖然這種方式有助於學生學習撰寫程式的技巧，但它也可能會造成剛入門的新手因為不熟悉語法和邏輯而受到挫折和焦慮。這樣的情況會影響學生的學習動機和自我效能感(self-efficacy)，進而導致學生降低參與度或甚至放棄學習(Haynes & Ericson, 2021)。

相對的，帕森斯問題能避免這樣的狀況，它讓學生在一個有趣的謎題中學習程式設計，透過排列程式碼塊、辨別正確的程式順序，讓學生逐漸掌握正確的語法和邏輯觀念，並最大限度的提高學生的參與程度，降低學生在學習途中放棄的

可能(Parsons & Haden, 2006)。此外，帕森斯問題提供即時回饋，幫助學生及時發現和解決自己的問題，彌補教師人員無法即時提供回饋的不足。在程式教育課程中，學生參與程度對於程式教育的成敗有著顯著的影響(Du et al., 2020)，帕森斯問題提供的即時回饋能夠改善傳統程式設計時無法得到即時回饋的窘境。

## 二、生成式人工智慧

生成式人工智慧(Generative Artificial Intelligence, GAI)是指一種人工智慧技術，可以根據輸入的訓練數據生成新的、與訓練數據相似但不同於訓練數據的內容，GAI的特點是能夠自主地生成全新的內容，而不是像其他人工智慧技術那樣僅能基於現有的數據進行推斷或分類。GAI可以根據給定的條件或上下文生成全新的、符合邏輯和內容要求的內容，因此在自然語言處理、圖像生成、音樂生成等領域中得到了廣泛應用。然而，GAI也存在一些問題和挑戰。例如，GAI可能會生成不合適或不恰當的內容，或者重復現有的內容，因此需要一定的後續處理和篩選，這需要使用者具有相應的能力。此外，GAI需要大量的訓練數據和計算資源，因此對於一些應用場景而言可能不太實際或不可行。

在程式教育方面，以教師端來說，GAI能提供更多元化和有效的教學方式，進而提高學生學習的效果，例如可以使用GAI來開發教材，並透過它來進行自動化的回饋和評估，快速的掌握學生學習進度和情況，從而節省時間和資源(Baidoo-Anu & Owusu Ansah, 2023)。以學生端來說，GAI的自動化回饋還可以幫助學生更好地理解抽象概念和過程，因為它能夠提供更具體的範例和說明，此外，它也可以提供不同的程式範例、解決方案或建議，讓學生參考和學習。GAI也可以幫助學生發揮創意，嘗試不同的程式風格或功能，並彌補教師無法即時回復學生問題的狀況，使問題能快速解決，進而增加學生學習動機。

如今各大公司都推出自己的GAI模型，如Open AI的ChatGPT、Microsoft的Bing Chat、Google的Bard等，其中ChatGPT是目前最熱門的GAI，不但能生成更自然的語言而且不限對話長度與數量，還能保留聊天紀錄。雖然Bing Chat與ChatGPT都是使用同一種語言模型訓練的，但與ChatGPT相比，Bing Chat是著重在搜尋網路資料相關領域，而目前也有對話數量的限制，也不能保留與之的聊天紀錄，可用性還不像ChatGPT般廣泛，而Bard截止目前也尚未對台灣地區開放，基於上述理由，本研究內容主要會採取ChatGPT作為主要研究工具。

## 參、研究實施與設計

### 一、研究工具

- 生成式人工智慧-ChatGPT

ChatGPT是由OpenAI所開發的聊天型機器人，它基於GPT-3.5架構訓練的大型語言模型，是一種生成式人工智慧技術。它可以透過學習大量的數據，自動

生成自然語言對話，使用者只要使用一般日常的對話模式就能與之對談。ChatGPT 擁有龐大的語言知識庫，能夠處理多種自然語言任務，例如對話生成、語言翻譯等，此外它也具有編寫和除錯電腦程式之能力。在實際學習場景應用中，ChatGPT 可以用於各種需要自然語言處理的課程等多個領域，如電腦科學和人工智慧等。在學生學習面向可以利用 ChatGPT 回答問題，以釐清不清楚的程式語法或邏輯概念概念，以教師教學端，則可以建構基本的學習內容框架，幫助教師更有效率的建立教材，或是依據學生程度製作考題和練習題(Hwang & Chen, 2023)。

## 二、研究過程

### 2.1 研究架構

本研究採取設計本位研究法 (Design-Based Research, DBR)，DBR 是一種強調結合理論與實務的研究方法，其主要目的是建立理論、設計產品和改善實務三種層面，這種方法透過對研究內容進行反覆迭代、修改和評估，從而逐步優化產品的效果和功能，並對實際應用產生重大的影響。設計研究的結果不僅僅是關於教育技術，還包括對實際應用產生的影響的洞察和理解(Hoadley & Campos, 2022)。而相較於其他種研究方法，設計研究法更加強調系統性的設計過程，此外，設計研究也更注重設計方案的實施和評估，以測試假設和研究結果是否能夠轉化為實際應用(翁穎哲 & 譚克平，2008)。

此研究將會進行以下循環，如下圖 1，分為四個階段：

1. 期望：此階段將會預期 ChatGPT 會產出的結果，本研究目的為輔助教師運用 GAI 初步規劃帕森斯問題，而不需要從頭開始編寫，進而節省資源和時間。
2. 設計：本研究透過選擇合適的提示(prompt)與持續與 ChatGPT 對話，藉此實施設計的方案，以達到研究目的。
3. 評估：根據與 ChatGPT 對話的結果，評估應用在實際的可行性，檢測研究結果轉為實際應用的可行性，並思考如何改進與修正提示。
4. 反思：本階段為 DBR 其中重要的一環，透過反思、檢討與 ChatGPT 的對話結果，並根據修正與改進的結果，重新設計方案，形成循環的設計研究過程。

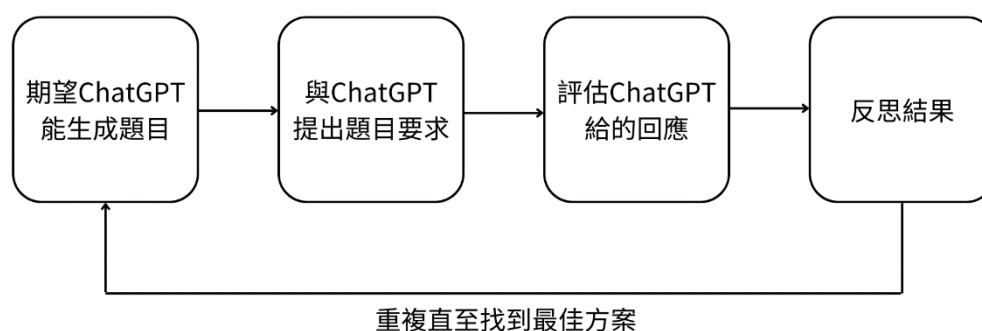


圖 1. 研究架構

## 2.2 無提供程式碼生成出帕森斯問題

本次研究透過與持續的與 GAI 對話來讓其生成之結果能達到預期的情況，也就是幫助教師生成帕森斯問題。首先本研究試問 ChatGPT 是否知道什麼是帕森斯問題，而能確實給出帕森斯問題的定義以及優缺點，這說明 ChatGPT 不會生成不相關的結果。

接著本研究試著問 ChatGPT 能否請他出一道帕森斯問題，而它雖然成功生成了一道帕森斯問題，但其結果並不如預期，其生成的程式碼雖然正確，但程式碼的長度太短，無發揮帕森斯問題的功能，如圖 2。



圖 2.請 ChatGPT 出一道帕森斯問題

接著再要求 ChatGPT 出另外一題帕森斯問題，但這次再要求的時候給出了明確的指示，例如明確指定要使用的程式語言以及想要出的核心觀念等，能夠讓 ChatGPT 更精確地回答問題，提高問題的實用性和適用性，相較於沒有指示問題時，ChatGPT 就可能會出現亂槍打鳥，不符合預期的情況產生。而此題令人眼之一亮之處在於，ChatGPT 提供了詳盡的題目描述，並附帶五段程式碼，其中一段為誘答選項，使得題目更加多元化和有趣，也更貼近帕森斯的核心概念，詳細對話內容如下所示。

**研究者：**請利用 Python 幫我出一個關於迴圈的 Parsons Problem。

**ChatGPT：**當然，以下是一個使用 Python 語言的 Parsons Problem，關於 for 迴圈的基礎概念和語法。

題目敘述：

給定一個列表 'numbers'，請依序完成以下操作：

1. 給定一個變數`total`，初始值為 0
2. 使用`for`迴圈將`numbers`中的元素逐一取出
3. 將每個元素加到`total`中
4. 在迴圈執行完畢後，輸出`total`的值

以下是五個可能的程式碼片段，請將它們排列成正確的順序，以完成題目要求。每個程式碼片段僅需使用一次，有一個片段不會被使用。

A. `for n in numbers :`

`total += n`

B. `total = 0`

C. `print(total)`

D. `numbers = [1, 2, 3, 4, 5]`

E. `total = sum(numbers)`

順序：D.B.A.C

然而，在實驗中發現 ChatGPT 生成的這道帕森斯問題存在一定的缺陷。具體來說，本研究接續要求 ChatGPT 打亂其中的三行程式碼，但 ChatGPT 卻把錯誤的選項也一起加入了帕森斯問題之中。這導致了題目的精確度和可靠性都有所下降，若教師未將其產生的結果更進一步檢查就使用，可能會對學生的學習效果產生了一定的影響。

**研究者：**請把解答中的程式碼只打亂其中 3 行

**ChatGPT：**題目敘述：

1. 給定一個列表 `numbers`，請依序完成以下的操作：
2. 設定一個變數 `total`，初始值為 0
3. 使用 `for` 迴圈將 `numbers` 中的元素逐一取出
4. 將每個元素加到 `total` 中
5. 在迴圈執行完畢後，輸出 `total` 的值

以下是五個可能的程式碼片段，請將它們排列成正確的順序，以完成題目要求。其中三個片段的順序已被打亂。每個程式碼片段僅需使用一次，有一個片段不會被使用。

A. `for n in numbers :`

`total += n`

B. `total = 0`

C. `print(total)`

D. `numbers = [1, 2, 3, 4, 5]`

E. `total = sum(numbers)`

順序：B. D. A. E. C.

由以上研究結果可以得知在使用 ChatGPT 時生成時並未提供明確的指示，或請它從無生成時，雖然 ChatGPT 還是會生成一段程式碼，但這時將使得 ChatGPT 產生錯誤的機會大幅提升，因此，再使用這項工具時進行生成程式碼或任何教學內容時，都應先事先規劃好架構，才能讓 ChatGPT 幫助工作。

## 2.2 提供完整程式碼生成帕森斯問題

上述是在無提供程式碼的情況下，要求 ChatGPT 生成帕森斯問題的狀況。然而，本研究也好奇當有一段完整的程式碼時，要求 ChatGPT 生成一道帕森斯問題的成效如何，本研究在此提供了一段使用 for 迴圈的程式碼用於計算成績的最高分數、最低分數、總分和平均分數，並詢問這段程式碼的意義並期望 ChatGPT 能夠理解整段程式碼的意義和功能，並解釋整個程式碼的運行過程。

**ChatGPT**：這段代碼用於計算使用者輸入的 10 個成績的最高分、最低分、總分和平均分，並輸出結果。

以下是代碼的執行過程：

1. 創建一個名為 'grade' 的空清單和一個名為 'total' 的整型變數，用於存儲成績和總分。
2. 使用 'for' 迴圈來反覆運算 10 次，每次從使用者輸入中獲取一個整數類型的成績，並將其添加到 'grade' 清單中。
3. 將新輸入的成績加到 'total' 中，用於計算總分。
4. 使用 'max' 函數和 'min' 函數來計算 'grade' 清單中的最高分和最低分。
5. 計算平均分並將結果存儲在 'average' 變數中。
6. 使用 'print' 函數輸出最高分、最低分、總分和平均分。

由上述回應得知 ChatGPT 確實知道這段程式碼的功能，接續本研究請求 ChatGPT 列出初學者在這段程式碼中最容易犯錯的觀念。根據 ChatGPT 的解釋，初學者在這段程式碼中最容易犯錯的觀念包括變數定義和使用的不一致、函數的錯誤使用以及控制流程的錯誤使用。

**研究者**：這段程式碼中初學者最容易出錯的觀念為何？

**ChatGPT**：學者最容易出錯的觀念是關於 Python 清單的索引和循環範圍的理解。

具體來說，以下是一些常見的錯誤：

1. 迴圈範圍錯誤：由於 Python 中的 range() 函數是從 0 開始的，因此對於這段代碼中的 range(0, 10)，實際上會反覆運算從 0 到 9，而不是從 1 到 10。因此，在列印出使用者輸入的每個成績時，應該使用 i+1 而不是 i。
2. 清單索引錯誤：當添加使用者輸入的成績時，應該使用 grade.append(num)

將新成績添加到清單的末尾，而不是使用 `grade[i] = num` 進行替換。因為在迴圈的第一次反覆運算時，清單 `grade` 是空的，使用 `grade[i]` 來引用清單中的元素會導致 `IndexError`。

3. 類型轉換錯誤：從使用者輸入中獲取的值是一個字元串類型，必須將其轉換為整數類型，否則計算總分時會導致 `TypeError`

因此，初學者需要注意在使用迴圈和清單時，正確理解迴圈範圍和清單索引的使用，以及注意將數據類型正確轉換。

為了幫助學習者更好地理解這些錯誤觀念，本研究接著利用 ChatGPT 生成帕森斯問題的形式來呈現這些錯誤。本研究要求 ChatGPT 將容易出錯的程式碼順序打亂，然而誘答選項是帕森斯問題的核心之一，誘答選項是一種與看似與其他程式碼相像，但實質上是不同的功用，也因此容易使學習者搞錯。然而本研究發現 ChatGPT 生成的帕森斯問題中並無誘答選項，因此期望使用 ChatGPT 產生另外一種具備誘答選項的帕森斯問題。

為了解決原本生成之程式碼並未具備誘答選項，我們要求 ChatGPT 新增幾行與原程式碼相似但錯誤的程式碼，並打亂程式碼順序，以形成含有誘答選項的帕森斯問題。

**研究者：** 在保留原有的程式碼下新增幾行相似但錯誤的程式碼，並且打亂程式碼順序

**ChatGPT：**

1. `average = sum(grade) / len(grade)`
2. `grade = []`
3. `for i in range(0,11):`  
    `num = int(input("請輸入第"+str(i+1)+"個數字："))`  
    `grade.append(num)`  
    `total += grade[i]`
4. `maxgrade = grade[-1]`
5. `minindex = grade.index(grade.min())`
6. `print("最低分的索引為：", minindex)`
7. `total = 0`
8. `print("請輸入 10 個數字，以空格分隔：")`
9. `input_str = input().strip()`
10. `input_list = input_str.split()`
11. `for i in range(0,10):`  
    `grade.append(int(input_list[i]))`  
    `total += grade[i]`
12. `print("總和成績為：",total)`

13. `average = total / len(grade)`

14. `print("平均成績為:", average)`

需要注意的是，這些程式碼可能會導致各種錯誤，例如：

1. 'average' 變數被重新賦值，而未重新計算，因此在代碼後面的計算中可能會出現錯誤。
2. 'range(0,11)' 將迴圈反覆運算 11 次，而不是 10 次，因此會添加一個額外的成績，並導致其他錯誤。
3. 'maxgrade' 的值被賦為了 `grade[-1]`，這意味著它是最後一個輸入的值，而不是最高分。
4. 'minindex' 的值是 `'grade.index(grade.min())'`，這不會計算出正確的最低分的索引，因為方法返回的是最低分的值，而不是其索引。
5. 在從使用者輸入的字串清單中提取成績時，代碼沒有檢查使用者是否已正確輸入 10 個值。這可能導致 'IndexError' 或 'ValueError'。
6. 'total' 變數在輸出最低分索引後被重新賦值為 0，而不是用於計算平均分，因此會導致計算錯誤。

為了修復這些錯誤，你需要重新排列代碼，將變數的值和類型恢復到正確的狀態，並修復任何語法錯誤。

ChatGPT 在解析中提供了更詳細的錯誤說明和解釋，以幫助學習者更好地理解錯誤觀念和如何修正它們。由與 ChatGPT 對話的結果得出，再經過多次修正提示詞(prompt)，可大幅的增加 ChatGPT 回應的準確性，而從結果也能發現相較請 ChatGPT 從無生成出一個帕森斯問題，在有給原使程式碼的情況下 ChatGPT 較不會出錯，還能依照使用者需求進行細部的更改，如在原有的程式碼多加幾行誘答的選項，此外，ChatGPT 有時會生成提示以外的東西，如在上方的一次對話中，ChatGPT 就自行生成题目的解析，並無其他提示，對於教學者來說可以再從對話中對教材進行修正，讓題目更臻完善，也能減少對於備課時間有限的壓力。

## 伍、結論與未來展望

因應生成式人工智慧的迅速發展，使用這項工具進行人機協作可謂不可抵擋之勢，若教師能熟悉的使用這項工具，它將能給予教師更多元、更彈性的方式製作教學或程式學習素材，提升教學品質，使設計的課程更具獨特性也更能貼近教學目標。然而生成式人工智慧依然還處在發展中的階段，因此還是會有產出不正確資訊之處，這時就需要教師重新審視其生成的內容，此外，使用這項技術時也須以邏輯性的方式敘述需求，才能發揮這項技術的優勢。透過與 GAI 的協作下，不管是前期的課程設計、抑或是課堂中的作業，都能使用這項工具來增進工作效率，減輕備課壓力。研究未來擬再進行更多使用 GAI 進程式教育方面的應用，以提升其對程式碼生成任務的準確性，以學生學習方面，使用 ChatGPT 作為學

生的學習輔助工具，如何應用 Chatgpt 給出提示而不是直接給答案，才不會失去學習的意義，此外，也可以針對使用 ChatGPT 進行學生評估，從而達到適性化教育，幫助學生在學習表現上能逐步提升。

## 參考文獻

- 翁穎哲 & 譚克平. (2008). 設計研究法簡介及其在教育研究的應用範例. 科學教育月刊, 307. [https://doi.org/10.6216/SEM.200804\\_\(307\).0003](https://doi.org/10.6216/SEM.200804_(307).0003)
- Baidoo-Anu, D., & Owusu Ansah, L. (2023). *Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning* (SSRN Scholarly Paper No. 4337484). <https://doi.org/10.2139/ssrn.4337484>
- Du, Y., Luxton-Reilly, A., & Denny, P. (2020). A Review of Research on Parsons Problems. *Proceedings of the Twenty-Second Australasian Computing Education Conference*, 195–202. <https://doi.org/10.1145/3373165.3373187>
- Haynes, C. C., & Ericson, B. J. (2021). Problem-Solving Efficiency and Cognitive Load for Adaptive Parsons Problems vs. Writing the Equivalent Code. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–15. <https://doi.org/10.1145/3411764.3445292>
- Helminen, J., Ihantola, P., Karavirta, V., & Malmi, L. (2012). How do students solve parsons programming problems? An analysis of interaction traces. *Proceedings of the Ninth Annual International Conference on International Computing Education Research*, 119–126. <https://doi.org/10.1145/2361276.2361300>
- Hoadley, C., & Campos, F. C. (2022). Design-based research: What it is and why it matters to studying online learning. *Educational Psychologist*, 57(3), 207–220. <https://doi.org/10.1080/00461520.2022.2079128>
- Hwang, G.-J., & Chen, N.-S. (2023). Editorial Position Paper: Exploring the Potential of Generative Artificial Intelligence in Education: Applications, Challenges, and Future Research Directions. *Educational Technology & Society*, 26(2). <https://www.jstor.org/stable/48720991>
- Parsons, D., & Haden, P. (2006). Parson's programming puzzles: A fun and effective learning tool for first programming courses. *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*, 157–163.