

生成式 AI 融入探究式學習對程式設計學習成效之影響：

以國二 Python 課程為例

The Impact of Integrating Generative AI into Inquiry-Based Learning on Middle School Students' Programming Performance: A Case Study of an 8th-Grade Python Curriculum

關宇亨 張循鏗

KUAN YU HENG CHANG HSUN LI

國立臺北教育大學 課程與教學傳播科技研究所 研究生

National Taipei University of Education Graduate School of Curriculum and
Instructional Communication Technology Student

E-mail : g111244009@grad.ntue.edu.tw

國立臺北教育大學 課程與教傳播科技學研究所 助理教授

National Taipei University of Education Graduate School of Curriculum and
Instructional Communication Technology Assistant Professor

E-mail : hsunli@tea.ntue.edu.tw

摘要

隨著科技的快速發展，程式教育的重要性在全球持續提升。然而，在學習程式的過程中，學生常面臨語法、概念等知識的挑戰，或缺乏有效的程式規劃與除錯能力。這些困難容易在學習時導致挫折感，進而影響學習成效。因此，如何運用有效的輔助工具及教學策略，使學生能夠有系統地學習程式語言，成為一項值得深入探討的議題。

本研究旨在探討將生成式人工智慧（Generative Artificial Intelligence, GAI）作為教學輔助工具，結合探究式學習策略，應用於程式語言教學中，對學生學習成效之影響。為達成此目標，本研究設計一套教學方案，運用 Gemini 作為 AI 輔助工具，並融入探究式學習的 POEE 架構，透過「預測、觀察、解釋、評估」四個步驟，引導國中學生學習 Python 程式語言，進一步分析其學習成效變化。

研究方法採用前實驗設計的單組前後測，對象為新北市某完全中學國二學生。透過 Python 學習成就測驗進行前後測，並將收集到的數據進行分析，預期能提升學生的 Python 學習成效。

關鍵字：生成式 AI、程式設計、探究式學習

Abstract

With the rapid advancement of technology, the importance of programming education continues to grow worldwide. However, students often encounter challenges related to syntax, conceptual understanding, and a lack of effective skills in planning and debugging programs. These difficulties can lead to frustration during the learning process, ultimately impacting learning outcomes. Therefore, identifying effective support tools and instructional strategies to help students systematically learn programming languages has become a critical issue worthy of further investigation.

This study aims to explore the impact of using Generative Artificial Intelligence (GAI) as a teaching support tool, integrated with an inquiry-based learning strategy, on students' programming learning outcomes. To achieve this goal, a teaching program was designed that incorporates Gemini as the AI assistant and adopts the POEE (Predict–Observe–Explain–Evaluate) model of inquiry-based learning. Through the four-step process of prediction, observation, explanation, and evaluation, middle school students were guided to learn the Python programming language, and their learning outcomes were analyzed.

A one-group pretest–posttest experimental design was employed. The participants were 8th-grade students from a public secondary school in New Taipei city. A Python learning achievement test was administered before and after the intervention, and the collected data were analyzed to examine changes in student performance. The results are expected to demonstrate improvements in students' Python programming achievement.

Keywords: Generative AI、Inquiry-based learning、Programming

壹、前言

隨著人工智慧與數位科技的快速發展，程式設計教育在全球教育體系中的重要性日益提升，各國政府普遍將其納入中小學課程(蔡進雄, 2019)。近 30 個國家，包括英國、美國、法國、韓國等，皆制定了程式教育相關政策(施又瑀, 2018)。而我國於 108 課綱中同樣將程式教育與運算思維納入資訊課程，彰顯出程式教育在國際教育體系中的關鍵地位。

然而，學生在學習程式語言時常面臨語法錯誤、邏輯概念誤解及缺乏規劃與除錯策略等多重挑戰(Qian & Lehman, 2017)，進而導致學習挫折並影響成效。同時，教師在「一對多」教學場域中，難以及時提供個別化指導，

尤其在初學者階段，學習需求差異化明顯，教學資源配置面臨挑戰。上述現象顯示，程式教育需要有效的輔助工具與教學策略，以提升學生學習品質並減輕教師負擔。

生成式人工智慧（Generative Artificial Intelligence, GAI）自 2022 年 OpenAI 推出 ChatGPT 後迅速發展，隨後如 Gemini、Claude 等模型相繼問世，廣泛應用於程式設計輔助、語言生成與邏輯推理等教育場域(Hu, 2023)。相關研究指出，生成式 AI 在提供即時回饋與差異化支持方面具備顯著優勢，能有效提升學生的學習動機與學習成效（Yilmaz & Yilmaz, 2023；Surameery & Shakor, 2023；Sun & Zhu, 2024），並促進自主學習與個別化進程（Cooper, 2023）。我國教育部自《AI 教育 × 教育 AI》政策發布以來，也持續將生成式 AI 的應用納入中小學科技課綱，展現對生成式 AI 應用於教育的支持。

探究式學習（Inquiry-Based Learning, IBL）是一種建構主義導向的教學策略，透過引導學生預測（Predict）、觀察（Observe）、解釋（Explain）與評估（Evaluate）四步驟，促進其深度思考與知識建構（Al Mamun, Lawrie & Wright, 2020；Urdanivia Alarcon et al., 2023）。研究顯示，將 IBL 應用於程式教學，可增進學生的問題解決能力與合作學習成效（Antonio & Prudente, 2024；林菁，2018；時文中等，2022）。

基於生成式 AI 在即時輔助與差異化教學支持上的潛力，以及 IBL 在系統化學習流程中的優勢，本研究提出結合生成式 AI 與 POEE 教學架構的創新模式，並設計以 Gemini 為 AI 輔助工具的 Python 教學方案。研究採用單組前後測實驗設計，對象為臺北市某完全中學國二學生，透過前後測資料分析，探討此教學模式對學生 Python 學習成效的影響，期望為程式教育提供具體實證與教學參考。故本研究之目的有三：

1. 設計一套結合探究式學習並將生成式 AI 導入程式教學的教學方案
2. 探討學生透過此教學方案對學生學習程式語言學習成效之影響
3. 探討學生透過此教學方案對學生高階思維能力之影響

貳、文獻探討

一、程式教育

（一）程式教育的重要性

程式設計教育的核心在於培養學生的運算思維（Computational Thinking, CT）。Wing（2006）所發表的論文研究中強調，運算思維是一種認知過程，涉及以電腦科學家的方式進行問題解決與系統性推理，其內涵遠超過單純的程式編寫技巧，而是一種可轉移的思考技能。

Koyuncu & Koyuncu（2019）指出，在高度資訊化的社會中，現代公民需具備能夠使用並生產資訊的能力，而程式設計正是一項培養這種能力

的關鍵工具。他們認為，程式設計是一個透過命令集進行電腦操作、與機器互動、解決問題並實現目標的過程，過程中涵蓋了邏輯建構、語意掌握與資訊組織等多重能力。因此，程式設計已被視為 21 世紀的關鍵技能之一，如何培養則有賴於程式教育。

而在實務層面，Brennan & Resnick (2012) 將運算思維拆解為三個向度，分別為：運算概念（如順序、迴圈、事件、條件判斷、運算符與資料結構等）、運算實踐（如迭測試除錯、模組化設計與抽象化等）以及運算視角（如自我表達、社群連結與批判思維等），突顯出程式學習在知識、實作與價值觀層面的多元意涵。而 Anderson (2016) 則從問題解決的角度，將計算思維具體歸納為五個核心步驟：分解問題、模式識別、抽象化、演算法設計以及解法評估，顯示程式設計不僅是技術操作，更是一種有效的知識處理框架。

綜上所述，程式教育的重要性體現在其對學生多層次能力的培養。能促進運算思維的形成，培養學生以邏輯且系統化的方式理解與解決問題 (Brennan & Resnick, 2012; Anderson, 2016); 同時發展抽象、分解、模式識別與演算法設計等高階思考能力 (Anderson, 2016; Koyuncu & Koyuncu, 2019)。隨著各行各業對程式能力的需求日增，從小培養學生的程式素養，不僅有助於未來職涯發展，更是因應科技時代不可或缺的核心素養。

(二) 程式教育面臨的挑戰

雖然程式學習在現今社會中，是一項相當值得培養的關鍵技能，但在實際進行教學時仍面臨許多挑戰。像是 Qian & Lehman (2017) 就曾提出程式設計的新手在剛開始接觸文字形程式時，會有三種常見的知識誤解。包括語法知識、概念知識以及策略知識。

首先在語法知識的部分，Altadmri & Brown (2015) 曾透過數據分析發現學生最常見的錯誤就是括號、方括號或引號不符。另外 Jackson et al. (2005) 也提到許多新手會有的其他錯誤像是沒有事先設定好變數、忘記在程式結束後加上分號以及混淆賦值運算子以及比較運算子的使用等等，這些都是不熟悉程式語言的語法所導致的學習問題。

除了語法知識外，概念知識也是學生在學習過程中容易遇到的問題，學生若對某些程式相關名詞的概念產生誤解，就會對程式學習產生障礙。(Bayman & Mayer 1983; Canas et al. 1994)。像是條件判斷或是迴圈等程式的概念也就容易讓學生混淆。例如有些學生認為，條件判斷式的 if 和 else 區塊中的程式都會被執行 (Sirchia, 2012)。有些學生則常常無法理解迴圈重複的範圍、哪些程式會被重複、回圈內的程式碼會被執行多少次等等。(Sleeman et al., 1986) 另外，許多程式語言中都存在 for 迴圈和 while 迴圈，但也有些學生通常會根據自己對某種迴圈的

熟悉程度，偏好使用自己熟悉的迴圈，而不是針對問題，使用該問題適合的迴圈結構。(Qian, Y. and Lehman, J., 2017)

最後，學生在學習程式時，也會因為缺乏策略知識而導致無法學好程式設計。所謂的策略知識包含了規劃、撰寫與除錯程式，並運用上述的語法和概念知識來解決問題。(McGill and Volet, 1997) 例如學生在學習程式初期常常難以理解任務並將問題拆解成較小的部分 (Muller, 2005; Robins et al., 2006)，或者無法將應該一起應用的程式區塊正確合併 (de Raadt, 2008)，以及缺乏對整體程式碼的理解並找到錯誤的所在 (McCauley et al., 2008)。

綜合來說，以上都是學生在學習程式時，在語法、策略以及概念這三個角度會遇到的問題與挑戰。針對這些問題，Fleury (1991) 有提到好的程式範例可以幫助學生建立對程式碼的理解，De-La-Fuente-Valentín 等人 (2013) 也指出提供即時的個人化回饋，可以預防性的提示學生可能會發生的錯誤，並增強學生的學習體驗。這些都是生成式 AI 可以協助完成的工作。因此本研究的其中一項研究目的，就是希望設計一套能導入生程式 AI 的教案，提高學生學習程式的學習成效。

二、 探究式學習

(一) 何謂探究式學習

探究式學習 (Inquiry-Based Learning, IBL) 是一種以學生為中心並源自於建構主義的教學方法。在探究式學習中，學生可以透過解釋、預測和實驗等步驟，針對以科學為基礎的主題和問題提出自己的問題，並做出決定，從而有機會進行探究 (Urdanivia Alarcon et al., 2023)。

不過，探究式學習的架構並非固定，像是 White 與 Gunstone (1992) 就提出過 POE 的探究模型，透過預測 (Predict)、觀察 (Observe) 以及解釋 (Explain)，幫助學生推理科學概念。但在他們之後，亦有其他學者例如 Hong (2019) 提出的 POQE 模型，加入評量 (Quiz)，用以評估學生的學習成效；Chen (2022) 則提出 POED 模型，增加了實作 (Do)，讓學生可以將概念與實體操作的過程做出對應的連結。

而本研究將採用 Al Mamun 與 Lawrie (2022) 所提出之 POEE 框架，在最後的步驟加上評估 (Evaluate)，並透過生成式 AI 為學生提供更快速、方便的回饋訊息，並進一步促進學生對知識的理解與建構 (李品慧、黃悅民，2024)。

(二) 探究式學習與 POEE 架構如何融入 AI 幫助學生學習程式

本研究會使用探究式學習，並以 Al Mamun 與 Lawrie (2022) 所提出之 POEE 架構作為教學策略的原因有二。

首先，邱柏升與鍾華翎（2021）認為程式教育的教學應避免傳統的講述式教學，否則容易導致學生的思考僵化。相反地，若採用探究式學習的策略，就能提升學生的程式知識以及高階思維能力。另外，時文中（2022）曾提到，探究式學習可有效提升程式設計的學習成效，且學生在學習程式設計的過程中，可以透過提問以及提示等方式減少學習過程中遭遇的挫折感，形成顯著的學習效果。由此可見，探究式的學習架構對於程式語言的學習具有正面的影響。

再者，Al Mamun 與 Lawrie（2022）所提出之 POEE 架構，相較於原始版本的 POE 教學，多了評估的環節，而評估的主要目的就包括提供學生即時的回饋以及彌補老師與同學支持的不足（Al Mamun & Lawrie, 2022）而這兩項恰好就是生成式 AI 擅長的地方，例如 Google Gemini 就能在講解程式設計的複雜概念時，提出詳細的範例與說明，即時告知學生具體的解決方案。（陳慧珊，2024）

綜上所述，探究式架構 POEE，除了適合程式教育之外，也因為其多了評估的步驟，所以能夠將生成式 AI 加以融合，不只讓學生能更個別化的了解自身遇到的問題與學習障礙，也能減少教師如打地鼠般，疲於奔命解決學生問題的窘境。

三、生成式 AI

（一）生成式 AI 在教育中的應用

隨著生成式 AI 的快速發展，其在教育領域的應用已成為研究關注的焦點。特別是在程式設計教育方面，ChatGPT 等生成式 AI 已被證明能夠提升學習效率、增強學生的理解能力，並改善整體學習體驗

（Yilmaz & Yilmaz, 2023; Sun et al., 2024）。透過提供程式碼建議、除錯輔助與即時反饋，生成式 AI 正在重塑學習方式，使學生能夠更靈活地獲取知識（Som Biswas, 2023; Surameery & Shakor, 2023）。

研究發現，生成式 AI 的輔助能有效提升學生的運算思維、程式設計自我效能及學習動機。透過生程式 AI 學生能更快理解複雜的程式設計概念，並在解決問題時展現更強的創造力與邏輯思維（Yilmaz & Yilmaz, 2023）。

同時，除錯是程式設計學習中最具挑戰性的部分，而生成式 AI 能有效幫助學生識別、理解並修正程式錯誤（Surameery & Shakor, 2023）。還能在學習過程中提供全方位的程式設計輔助，包括程式碼補全、錯誤檢查與文件生成等功能（Som Biswas, 2023）。透過這些功能，學生能夠更快掌握程式語法，減少因語法錯誤或概念不清而造成的學習障礙，從而提升程式設計技能。

這種即時且互動的學習方式不僅減少了學生對傳統指導的依賴，也讓他們能夠更加主動地探索與實踐程式設計技能。在這樣的學習環境

下，學生能夠快速從錯誤中學習，進一步提升程式開發的能力。此外，生成式 AI 的即時反饋與互動性質使學生能夠更快適應並掌握新知識，進一步增強學習動機與自信心 (Sun et al., 2024)。

綜合來看，透過生成式 AI 的幫助，學生不僅能在學習上獲得更大自主性，還能培養主動學習的態度，這對於長期的學習成長至關重要。生成式 AI 在教育領域展現出極大的應用價值，尤其是在程式設計學習方面。它能夠提升學習成效、增強學生的除錯能力、改善學習體驗，並作為教學的有力補充工具。然而，為了發揮生成式 AI 的最大效用，仍需謹慎評估其局限性，並確保它被合理地整合到教學實踐中 (Yilmaz & Yilmaz, 2023; Sun et al., 2024)。隨著生成式 AI 的進步，如何將其與傳統教育方式結合，以創造更加高效的學習環境，將是未來教育研究的重要課題。

(二) 生成式 AI 在教育中需解決的問題

儘管生成式 AI 在教育領域的應用越來越廣泛，尤其是在程式設計教學方面，許多研究探討了生成式 AI 對學生學習的影響，也都發現生成式 AI 能提升學生的運算思維、學習動機和程式設計能力，但教育者仍需關注其潛在的挑戰與問題，以確保技術能夠被正確利用。

其中，生成式 AI 核心問題之一是其提供資訊的準確性與可靠性。在程式設計教育中，生成式 AI 可用於程式碼建議、除錯與解釋，但其輸出並非總是正確的。研究顯示，學生在使用生成式 AI 輔助學習時，可能會遇到不準確的資訊、有限的輸入與輸出以及技術穩定性問題 (Sun et al., 2024)。此外，生成式 AI 的表現依賴於其訓練數據的品質，若數據含有錯誤或偏見，可能會影響其除錯與解釋的準確度 (Surameery & Shakor, 2023)。因此，在教育應用中，教師與學生應該謹慎驗證生成式 AI 的輸出，而非完全依賴生成式 AI 工具。

另外，生成式 AI 的便捷性可能會影響學生的自主學習能力與批判性思維。研究指出，在生成式 AI 輔助模式下，學生傾向於花費較少時間在除錯過程中，並更頻繁地依賴生成式 AI 提供回饋 (Sun et al., 2024)。雖然這有助於提升程式設計能力，但也可能降低學生獨立思考與解決問題的能力。此外，若學生過度依賴生成式 AI 提供答案，可能會削弱他們對程式邏輯與概念的理解。因此，教育者應鼓勵學生在使用生成式 AI 時，仍需保持主動探索與思考，以確保學習的深度與效果。

生成式 AI 對學生的學習動機與態度也可能產生影響。研究顯示，使用生成式 AI 的學生在運算思維與程式設計自我效能方面的得分顯著高於未使用生成式 AI 的學生，顯示生成式 AI 可能增強學習動機 (Yilmaz & Yilmaz, 2023)。然而，過度依賴生成式 AI 可能導致學生缺乏主動學習的動力，影響長期的學習成果。因此，教育者應在課堂中適

當規劃生成式 AI 的使用方式，確保其能夠輔助學習而非取代學生的思考過程。

最後，生成式 AI 在教育領域的應用具有潛力，但也伴隨著一系列挑戰。準確性問題、學習自主性的影響、與傳統學習工具的整合，以及學生學習動機的變化，都是在教育應用中需要關注的議題。為了最大化生成式 AI 的優勢並降低潛在風險，教育者應該謹慎設計生成式 AI 的應用方式，鼓勵學生發展批判性思維，並結合多種學習資源來提升學習效果。

參、研究實施與設計

一、研究設計

本研究採用前實驗研究設計的單組前後測設計，對象維新北市某完全中學之其中一個班級，針對 40 位八年級學生先進行 Python 學習成就測驗以及高階思維能力量表前測，接者進行 5 週 10 堂以探究式學習結合生成式 AI 之 Python 教學，並於課程結束後，施予 Python 學習成就測驗以及高階思維能力量表後測。最後透過取得之成績進行數據分析，探討探究式學習結合生成式 AI 的程式教學對學生學習成效與高階思維能力之影響。

二、研究工具

(一) 探究式學習導入生成式 AI 之 Python 教學方案

本研究之探究式學習導入生成式 AI 之 Python 教學方案內容是將 POEE 的四個步驟融入 Python 的教學之中，讓學生在學習 Python 的基礎知識，如變數、條件判斷、迴圈等概念時，有一個固定的架構與步驟可以正確地與生成式 AI 對話，並獲得其協助。

(二) Google Gemini AI

本研究將採用 Google 之 Gemini 作為本研究之生成式 AI，原因為陳慧珊（2024）之研究中有提到，Gemini 相較於 ChatGPT 對於程式設計的概念與複雜結構的說明，能提供更好的示範與解釋，讓學習者克服學習障礙。

(三) Python 測驗卷

本研究採用陳慧珊（2024）編製之 Python 測驗卷，以探討此教學方案是否能確實幫助學生提升程式的學習成效。該測驗卷之題目皆出自翰林雲端命題大師，題目由系統區分難中易三個等級，分別在評量中各取出題目偏難一題、題目偏中四題及題目偏易五題，共十題測驗，將於前後測調換題目順序，且不公布前測成績，避免學生背下題目與答案。

(四) 高階思維能力問卷

本研究採用李品慧、黃悅民（2024）所使用之高階思維能力量表，其量表改編自 Chai、Deng（2015）之研究，將其研究中的創造力、問題解決能力以及批判性思考能力提出，進行翻譯後編制為高階思維能力量表，並在信度分析後得出，所有信度皆高於 0.7。

三、資料處理與分析

本研究將在課程開始前以及結束後分別以 Python 測驗卷以及高階思維能力量表實施前、後測，在收集完測驗結果後，以 spss25 進行相依樣本 t 檢定，透過結果檢視學生的學習成就和高階思維能力在此教學方案下是否有顯著提升。

肆、結果與討論

本研究以國中二年級學生為教學對象，採用探究式學習理論結合生成式 AI，融入程式設計教學之中。為檢視教學介入成效，於課程實施前後分別進程式學習成效及高階思維能力之前後測。目前研究資料尚在整理與分析中，依據研究假設，預期結果可能包括：（1）探究式學習結合生成式 AI 能有效提升學生在程式學習上的學習成效；（2）探究式學習結合生成式 AI 能促進學生高階思維能力的發展。透過本研究的結果，期待能為未來程式教育之教學設計提供實證基礎，並對生成式 AI 在教育情境中之互動與協作應用，提供具體建議與參考方向。

五、未來展望

探究式學習結合生成式 AI 於程式教育領域的未來展望相當廣泛。例如：未來可進一步開發個性化的 AI 家教輔助學生學習，以因應學生不同的學習風格與需求，提升學習成效，甚至亦可應用於特殊教育領域，協助學習有困難或需要額外支持的學生發展程式設計能力。此外，隨著生成式 AI 技術持續進步，其在教學過程中的互動性、回饋即時性與問題解決引導等功能將更為精緻，有助於培養學生的高階思維與自主學習能力。未來亦可透過跨領域合作，例如教育學、人工智慧與認知科學的整合，推動探究式學習與 AI 輔助教學模式的創新與發展。探究式學習結合生成式 AI 在程式教育的應用，預期將為學生的學習歷程、問題解決能力及未來數位素養的養成，帶來深遠的影響與益處。

參考文獻

一、中文部分

邱柏升、鍾華栩（2021）。國中小學實施程式教育之問題與建議。《台灣教育研究期刊》，2（1），325-334。

- 李品慧、黃悅民 (2024)。探究 GPT：強化 ChatGPT 以促進 STEM 教育中的探究式學習。載於社團法人台灣工程教育與管理學會 (主編)，**工程、技術與 STEM 教育研討會** (頁 218-231)。社團法人台灣工程教育與管理學會。https://doi-org.metalib.lib.ntue.edu.tw/10.29619/STEM.202408.0018
- 林菁 (2018)。國小探究式資訊素養融入課程之研究：理論與實踐。**教育資料與圖書館學**，55(2)，103-137。
https://doi.org/10.6120/JoEMLS.201807_55(2).0004.RS.CM
- 施又瑀 (2018)。臺灣程式教育的困境與展望。**臺灣教育評論月刊**，7 (9)，1-8。
- 時文中、廖悅淳、洪梓瑜 (2022)。結合遊戲軟體情境之探究式學習法：程式學習之個案研究。**數位學習科技期刊**，14 (2)，1-27。
- 陳慧珊 (2024)。結合 ARCS 動機模型與 AI 生成工具以增進程式學習成就之研究-以國中 Python 課程為例。〔碩士論文。國立雲林科技大學〕臺灣博碩士論文知識加值系統。
- 蔡進雄 (2019)。未來教育新趨勢—各國程式設計教育的動態。**國家教育研究院電子報第 181 期**

二、英文部分

- Al Mamun, M. A., Lawrie, G., & Wright, T. (2020). Instructional design of scaffolded online learning modules for self-directed and inquiry-based learning environments. *Computers & Education*, 144, 103695.
- Altadmri, A., & Brown, N. C. (2015, February). 37 million compilations: Investigating novice programming mistakes in large-scale student data. *In Proceedings of the 46th ACM technical symposium on computer science education* (pp. 522-527).
- Anderson, N. D. (2016). A call for computational thinking in undergraduate psychology. *Psychology Learning & Teaching*, 15(3), 226-234.
- Antonio, R. P., & Prudente, M. S. (2024). Effects of Inquiry-Based Approaches on Students' Higher-Order Thinking Skills in Science: A Meta-Analysis. *International Journal of Education in Mathematics, Science and Technology*, 12(1), 251-281.
- Bayman, P., & Mayer, R. E. (1983). A diagnosis of beginning programmers' misconceptions of BASIC programming statements. *Communications of the ACM*, 26(9), 677-679.
- Biswas, S. S. (2023). Role of chat gpt in public health. *Annals of biomedical engineering*, 51(5), 868-869.

- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. *In Proceedings of the 2012 annual meeting of the American educational research association*, Vancouver, Canada (Vol. 1, p. 25).
- Canas, J. J., Bajo, M. T., & Gonzalvo, P. (1994). Mental models and computer programming. *International Journal of Human-Computer Studies*, 40(5), 795-811.
- Chai, C. S., Deng, F., Tsai, P. S., Koh, J. H. L., & Tsai, C. C. (2015). Assessing multidimensional students' perceptions of twenty-first-century learning practices. *Asia Pacific Education Review*, 16, 389-398.
- Chen, J. C. (2022). Developing a cycle-mode POED model and using scientific inquiry for a practice activity to improve students' learning motivation, learning performance, and hands-on ability. *Interactive Learning Environments*, 30(7), 1252-1264.
- Cooper, G. (2023). Examining science education in chatgpt: An exploratory study of generative artificial intelligence. *Journal of Science Education and Technology*, 32(3), 444-452
- De-La-Fuente-Valentín, L., Pardo, A., & Kloos, C. D. (2013). Addressing drop-out and sustained effort issues with large practical groups using an automated delivery and assessment system. *Computers & Education*, 61, 33-42.
- De Raadt, M. (2008). *Teaching programming strategies explicitly to novice programmers* (Doctoral dissertation, University of Southern Queensland).
- Fleury, A. E. (1991). Parameter passing: The rules the students construct. *ACM SIGCSE Bulletin*, 23(1), 283-286.
- Hong, J. C., Tsai, C. R., Hsiao, H. S., Chen, P. H., Chu, K. C., Gu, J., & Sitthiworachart, J. (2019). The effect of the "Prediction-observation-quiz-explanation" inquiry-based e-learning model on flow experience in green energy learning. *Computers & Education*, 133, 127-138.
- Hu, K. (2023). *ChatGPT Sets Record for Fastest-Growing User Base—Analyst Note*. <https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01>
- Jackson, J., Cobb, M., & Carver, C. (2005, October). Identifying top Java errors for novice programmers. *In Proceedings frontiers in education 35th annual conference (pp. T4C-T4C)*. IEEE.
- Koyuncu, A. G., & Koyuncu, B. (2019). The Universal Skill of 21st Century, Coding and Attitude of Secondary School Students towards Coding. *Language Teaching Research Quarterly*, 11, 68–80.

- McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: a review of the literature from an educational perspective. *Computer Science Education, 18*(2), 67-92.
- Muller, O. (2005, October). Pattern oriented instruction and the enhancement of analogical reasoning. *In Proceedings of the first international workshop on Computing education research* (pp. 57-67).
- Robins, A., Haden, P., & Garner, S. (2006, January). Problem distributions in a CS1 course. *In Proceedings of the 8th Australasian Conference on Computing Education-Volume 52* (pp. 165-173)..
- Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE), 18*(1), 1-24.
- Sirkiä, T. (2012). Recognizing programming misconceptions-An analysis of the data collected from the UUhistle program simulation tool.
- Sleeman, D., Putnam, R. T., Baxter, J., & Kuspa, L. (1986). Pascal and high school students: A study of errors. *Journal of Educational Computing Research, 2*(1), 5-23.
- Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education, 21*(1), 14.
- Surameery, N. M. S., & Shakor, M. Y. (2023). Use chat gpt to solve programming bugs. *International Journal of Information Technology and Computer Engineering, (31)*, 17-22.
- Urdanivia Alarcon, D. A., Talavera-Mendoza, F., Rucano Paucar, F. H., Cayani Caceres, K. S., & Machaca Viza, R. (2023, May). *Science and inquiry-based teaching and learning: a systematic review. In Frontiers in Education (Vol. 8, p. 1170487)*. Frontiers Media SA.
- White, R., & Gunstone, R. F. (1992). Prediction-observation-explanation. In R. White & R. F. Gunstone (Eds.), *Probing understanding* (pp. 44-64). London, UK: The Falmer Press.
- Wing, Jeannette M. "Computational thinking." *Communications of the ACM 49.3* (2006): 33-35.
- Yilmaz, R., & Yilmaz, F. G. K. (2023). Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning. *Computers in Human Behavior: Artificial Humans, 1*(2), 100005.