

教育部教學實踐研究計畫成果報告
Project Report for MOE Teaching Practice Research Program

計畫編號/Project Number：PBM1121652

學門專案分類/Division：商業及管理

計畫年度：112 年度一年期 111 年度多年期

執行期間/Funding Period：2023.08.01 – 2024.07.31

以「複製貼上」建構程式思維，行不行？
AI 思維與程式設計

計畫主持人(Principal Investigator)：王福星

協同主持人(Co-Principal Investigator)：

執行機構及系所：中國文化大學／資訊管理學系

成果報告公開日期：立即公開 延後公開

繳交報告日期(Report Submission Date)：2024 年 08 月 30 日

以「複製貼上」建構程式思維，行不行？

一. 本文 Content

1. 研究動機與目的 Research Motive and Purpose

教育部自 107 年以來鼓勵教師針對教學現場問題，提出教學實踐研究計畫案，也鼓勵教師做計畫成果分享，並遴選學門績優計畫。歷經 107-111 這 5 年的計畫推動，已經有為數不少教學實踐研究計畫案被執行，其個別計畫成果也經發表後，被置於平台上做進一步的擴大分享。有關程式課程的教學現場問題，被廣泛地提出。其中關鍵字有”程式”二字的計畫案共有 275 筆，分布於多個學門：[專案]大學社會責任(5 件)、[專案]技術實作(21 件)、人文藝術與設計(10 件)、工程(123 件)、民生(5 件)、生技醫護(1 件)、商業及管理(33 件)、教育(26 件)、通識(含體育)(40 件)、數理(11 件)。在教學實踐研究計畫成果交流平台上，教學者可以就自己遇到的教學現場問題，找到相關計畫案的教學實踐方式，再進一步客製化為自己的課程相應作法；或是從交流平台上之計畫的實施課程或對象，了解自己所教授課程可能存在的問題及其解決或改善的方法。本計畫所實施的課程與程式設計相關，實施的對象為非資訊領域的學生，交流平台上 107-111 的計畫案中，符合程式設計的計畫案有 275 筆，另符合非資訊領域的學生的計畫案有 23 筆，其中探討非資訊領域的學生的程式設計相關計畫案有 20 筆。另外，在 112 和 113 學年度關鍵字有”程式”二字的教學實踐研究計畫案分別有 69 件和 75 件，更為顯示程式課程的教學是廣受研究和討論的議題。

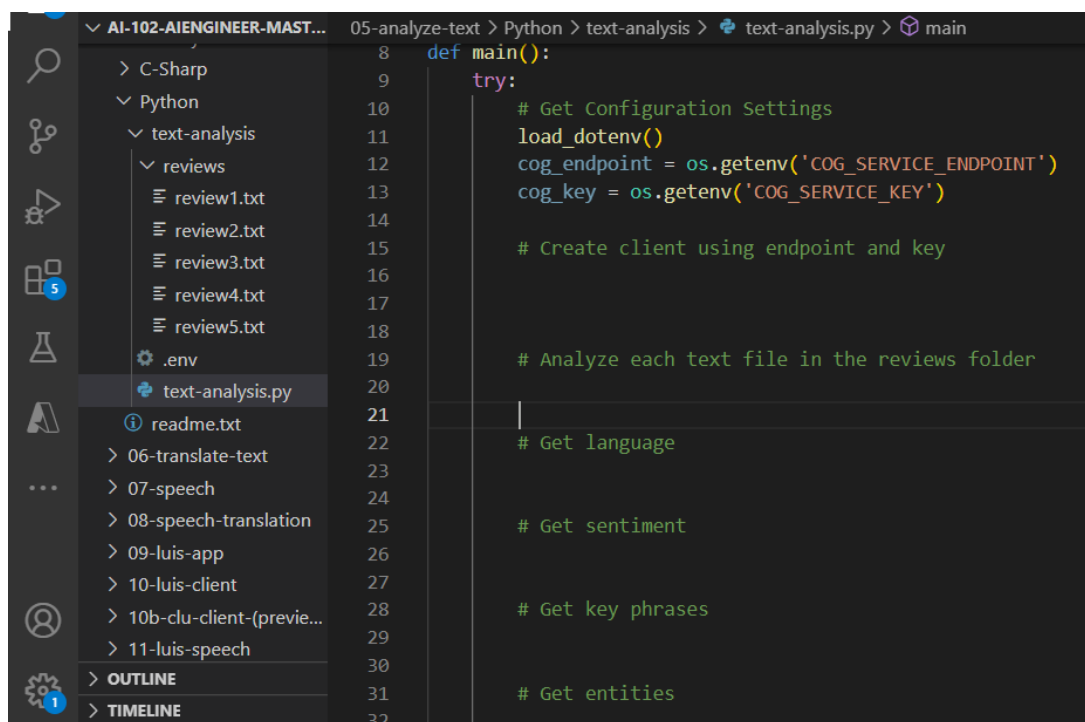
做好程式設計教育絕非易事，教非資訊相關科系學生寫程式，是否更是難上加難？申請人覺得未必。因應 AI 時代，教育部推動「大學程式設計教學計畫」，積極培育非資訊科系大學生也能具備運算思維與基礎程式設計的能力，以期能將程式概念與自身專業領域相結合，設計創新應用，提升個人

未來進入職場以及臺灣整體在數位經濟的關鍵競爭力。由此可知「運算思維」、「程式設計」兩者間能夠相輔相成，且都是目前非資訊科系大學生所亟待培養的能力。本計畫實施之選修課程「AI 思維與程式設計」，修課學生來自於各學院非資訊相關學系，修課人數不多，不過，學生的學習動機明顯，有機會導入自主學習。由於本計畫案實施的對象為非資訊背景學生，而且實施的課程為兩學分的單學期課，【學生難以在短時間內建構程式思維】是教學現場的第一個問題。運算思維的培養和程式設計的訓練是相輔相成，對於資訊相關科系的學生，可藉由離散數學、資料結構等基礎課程，強化其運算思維；然而，非資訊相關科系的學生要在一個學期內寫出程式，用傳統的教學方式難以達成。反思之下，本計畫提出一套因應的教學模式。

2. 研究問題 Research Question

非資訊領域的學生接觸程式設計課程，若沿用教授資訊領域學生的教學模式和教材，可以確定將造成學生的學習挫折與障礙，故必須設計適合教授非資訊領域學生的教學模式和教材。另外，由於課程時間僅有一學期，而且是兩學分的課，若要讓學生從無到有，製作出一個符合商管需求的完整程式，應該極不容易達成。因此，本計畫嘗試將就各教學單元欲完成的程式功能，先切割成數個獨立的子功能模組區塊，讓學生能夠按照欲達成的單元目標，以組合方式拼湊出符合邏輯順序的程式，據以培養學生的運算思維能力。要達成此一目的，教材的設計是成敗與否的重要關鍵。以申請人使用過的微軟 AI-102 實驗為例(圖一)，在做文本分析(text-analysis)時，欲分析出貼文是使用哪一國的語言(language)? 是正面或負面的貼文(sentiment)? 貼文包含哪些關鍵字(key phrases)? 文本分析是單元目標，而語言、語意、關鍵字等為子功能模組。學習者欲達成文本分析，需要釐清各子功能模組在完整程式中的順序，若有些模組的順序放得不對的話，或許只是影響到輸出結果的順序而已，但是也可能是造成邏輯上的問題，而產生不出結果，甚至出現錯誤而執

行不下去。故如何設計適合拼圖式的程式教材，是為了達成讓【學生在短時間內建構程式思維】此一目的重要的研究主題。



The image shows a code editor window with a dark theme. The left sidebar displays a file explorer with a tree view containing folders like 'C-Sharp', 'Python', and 'text-analysis', and files like 'review1.txt' through 'review5.txt', '.env', and 'text-analysis.py'. The main editor area shows the code for 'text-analysis.py' with line numbers 8 to 32. The code includes a 'def main()' function with a 'try:' block containing several commented-out lines: '# Get Configuration Settings', '# Create client using endpoint and key', '# Analyze each text file in the reviews folder', '# Get language', '# Get sentiment', '# Get key phrases', and '# Get entities'. The code uses 'os.getenv()' to retrieve environment variables for 'COG_SERVICE_ENDPOINT' and 'COG_SERVICE_KEY'.

圖一：微軟 AI-102 實驗 05-analysis-text 的 Python 程式畫面

由於課程時間實在有限，學生需要配合做好自主學習。翻轉教室的自主學習概念廣為流行，惟如何促成實質效益仍需要探討。申請人於 111 學年執行中的計畫案，即將翻轉教室應用於程式設計課程，探討教學模式對於學生的學習動機與成效。翻轉教室強調以學生學習為中心，相較於傳統教學，學生是否能於課前完成預習課程內容，是翻轉教室能否成功的重要關鍵。考慮到本計畫在教材設計上做了重大的改變，我們更重視學生是否釐清解決問題的邏輯順序，對於不同順序所造成的影響，學生是否有能力可以想得周全。因為程式的學習還是需要大量的演練，在學生做自主學習時，本計畫先不要求學生翻轉，反而是課後演練時，學生如何可以自我評量學習成效，或是能否有一個評量系統可以輔助。要製作一個可以評量中大型程式的評量系統並不容易，但是，若是針對如上述已經設計成區塊模組的程式，學生寫程式，相當於複製貼上區塊模組於適當位置，按如此的程式製作方法，計畫將嘗試提供一個評量工具，讓學生使用於自主學習的評量。於是【如何提供學生商

管程式的即時評量】將是本計畫的另一個研究主題。

(1).文獻探討 Literature Review

運算思維 (Computational thinking) 是一種以數學思維為核心的分析性思維，對於工程中的問題解決、設計和評估複雜系統非常有用。ISTE (2015) 指出，計算思維不會取代創造力、邏輯思維和決策能力，而是通過計算技術進一步提升這些技能。對於運算思維，學者和企業紛紛提出其對運算思維的定義，以及架構。2006 年，微軟研究院周以真副總裁認為運算思維是當我們碰到問題時，可以利用運算思維來理解、分析問題，並發展出可能的解決方式 (Wing, 2006)。Wing 提出運算思維的四個核心能力 (Wing, 2006)：

1. 拆解 (Decomposition)，將一個大問題或困難的問題，拆解成數個小問題或簡單的問題。
2. 模式識別 (Pattern recognition)，從不同的複雜問題之間，找出問題的相似點，即為模式。未來遇到其他複雜的問題時，可以用過去的模式解法，有效解決新的問題。
3. 抽象化 (Abstraction)，找出最主要導致此模式的原則或因素，目的是通過過濾非基本特徵並保留最相關的特徵來捕捉問題的本質，以創建簡化的表示或模型。
4. 演算法 (Algorithm)，制定一組精確的步驟，按特定順序以解決問題。

在過去十年，以程式設計提升學生的運算思維，受到了相當大的關注。許多研究檢驗了積木式程式設計語言 Scratch 在發展青少年運算思維方面的學習成效 (Sáez-López 等人, 2016; Pérez-Marín 等人, 2018; Sáez-López 等人, 2019)。許多其他作者將運算思維定義為一種技能，例如，Anderson 將運算思維視為計算機科學家在解決問題時的思考方式 (Anderson, 2016)。Gouws 將運算思維描述為一個概念，它從具體的計算機科學實踐中提取像計算機科學家一樣思考的思維過程，並提供對計算機科學家如何處理問題的更普遍的理解 (Gouws 等人, 2016)。雖然到目前為止，就如何教授運算思維技能或是否該將程式設計納入運算思維的教學並沒有達成一致性，Curzon 等人認為沒有程式設計的運算思維教學會缺乏創造性和挑戰性 (Curzon 等人, 2006)。程式設計語言可分為文字式程式設計 (text-based programming) 和積木式程式設計 (block-based programming)。一般而言，文字式程式設計程式符號較為複雜。積木式程式語言廣泛運用於各教育階段資訊課程，目前常見的積木式程式設計環境如由卡內基美隆大學推出的 Alice，以及由麻省理工學院推出的 Scratch。Brennan 對於運算思維提出了幾個要素，包括序列 (Sequences)、迴圈 (Loops)、平行 (Parallelism)、事件 (Events)、條件 (Conditionals)、運算元 (Operators)、資料 (Data) 等，而且認為 Scratch 可培養學生多種運算概念，並進一步轉化為運算思維能力 (Brennan & Resnick, 2012)。邱仁一針對積木式程式設計之學習成效做後設分析後發現，學生學習積木式程式設計，在程式設計能力、問題解決、運算思維和學習動機上皆有正向學習成效 (邱仁一、崔夢萍, 2021)。

非資訊領域的學生來自於不同的學習背景，需要藉由不同的學習誘因，以提

升學生學習動機(何素美, 2021)。何素美教授運用PBL問題導向教學模式, 結合悅趣化學習, 在提升學生學習程式設計之學習動機的目標下, 提升學生的邏輯思考能力(何素美, 2021)。本計畫也將採取PBL問題導向教學模式, 協助學生結合商管與自己的領域, 訂定問題, 藉由小組討論、實際解決問題的方式, 來達到自主學習的目的。

3. 教學設計與規劃 Teaching Planning

教學目標：

- 1) 培養學生在電腦科學領域中所需具備的基礎知識
- 2) 培養學生邏輯思考的能力。
- 3) 培養學生解決問題的能力。

教學方法：

- 1) **採用自行設計的區塊式程式教材的直接教學法**：著重知識技能的習得、理解與應用。區塊式程式教材係指將原配合教學單元所使用的 Python 程式, 切割成具有特定意涵的程式區塊(即片段程式)。
- 2) **練習教學法**：練習可以有效地形成和改進學生的技能, 發展學生的能力, 進一步開發學生的操作能力和創造能力。教師教導學生善用程式開發環境所提供的除錯工具, 除錯工具可以讓學生容易發現邏輯錯誤所在, 讓學生做有效率的練習。
- 3) **合作學習教學法**：各分組組員在作業練習時為學習共同體, 進行合作學習。平時測驗的作答, 由小組共同協力完成。
- 4) **自主學習法**：在適合的課程子單元主題融入問題導向設計, 繼續強化學生碰到問題的自我問題解決能力。本計畫將設計一個可以提供即時評量的聊天機器人, 可做為學生進行區塊式程式演練時的評量工具。

1. 學生成績考核與學習成效評量工具

教師依循教學目標設計成績考核方式(期中測驗評量 30%, 期末測驗評量 30%, 實做評量 40%)並事先公布, 讓學生清楚考核方式以及標準。本實施課程的評量著重於程式思維能力的評量, 評量分為兩部分, 1. 評量學生的實做能力, 2. 評量學生的情意面向。

實做能力部分包含基礎資訊技術能力和團隊合作以及敬業精神。下表為實做能力部分的考核方式以及標準。

表 2：實做能力的考核方式以及標準 (Rubrics 評量指標)

L2: 基礎資訊技術能力			
Objectives/Traits	Excellent	Satisfactory	Unsatisfactory

[L2T1]具備電腦科學領域所需基礎知識	可提供實例說明電腦科學領域應用	瞭解電腦科學領域之資訊	未對電腦科學領域有清楚認識
[L2T2]具備電腦程式設計與邏輯思考能力	可撰寫具有良好邏輯思考的電腦程式	瞭解電腦程式設計且具有邏輯思考能	未對電腦程式設計有清楚認識
L4: 團隊合作以及敬業精神			
Objectives/Traits	Excellent	Satisfactory	Unsatisfactory
[L4T1]瞭解團隊合作意涵	可以實例說明團隊合作意涵	瞭解團隊合作意涵	對團隊合作意涵無清楚之認識
[L4T2]願意為團體服務或貢獻	具有主動為團體服務或貢獻熱誠	有意願為團體服務或貢獻，但缺乏主動性	缺乏為團體服務或貢獻之意願
[L4T3]會傾聽與尊重他人不同之意見	會設身處地，納入或考量不同立場與意見	議題討論時，會嘗試瞭解他人想法或意見	議題討論時，只顧陳述自己的意見

自我效能評量部分，本研究採用 Korkmaz 等人所編製的運算思維量表，下表為題目：

表 3：自我效能量表(來源：Korkmaz et al., 2017)

編號	題目
1	我喜歡那些對自己的大多數決定都很確定的人。
2	我喜歡那些現實和中立的人。
3	我相信如果有足夠的時間和努力，我可以解決大部分面臨的問題。
4	我有一種信念，當我遇到新情況時，可以解決可能出現的問題。
5	在解決我的問題時，我相信自己能夠在制定計劃時應用它。
6	做夢讓我的重要項目浮出水面。
7	在尋找問題解決方案時，我相信自己的直覺和“正確”與“錯誤”的感覺。
8	當我遇到一個問題時，在轉向另一個主題之前，會停下來思考那個問題。
9	我可以立即建立解決問題的等式。
10	我認為我對數學過程有特別的興趣。
11	我認為我更容易理解用數學符號和概念進行的說明。
12	我相信我能輕易抓住數字之間的關係。
13	我可以用數學表達我在日常生活中遇到的問題的解決方法。
14	我可以將口頭表達的數學問題數字化。
15	我擅長制定解決複雜問題的計劃。
16	嘗試解決複雜問題很有趣。
17	我願意學習具有挑戰性的事物。
18	我為能夠精確思考而感到自豪。
19	在比較手頭的選項並做出決定時，我會使用系統方法。

20	我在心中演示問題解決方案時遇到困難。
21	我在解決問題時，不知道應該在哪裡以及如何使用變量如 X 和 Y。
22	我無法按計劃逐步應用解決方法。
23	在考慮問題的可能解決方法時，我無法產生太多選項。

4. 研究設計與執行方法 Research Methodology

本研究採用量化研究部分，主要在探討學生對教學的學習成效與滿意度。

本研究將透過本校教學意見期中問卷和期末問卷調查、運算思維問卷、Bebras 國際運算思維挑戰賽等量化設計進行資料收集。本研究使用套裝統計軟體進行量化分析，主要針對運算思維問卷量表，以李克特氏五分量表將資料做量化轉換。統計分析方法包括以平均數等敘述性的統計分析樣本的變項分析，再以描述性統計等統計方法進行數據分析。從學生端檢視教學品質滿意度，到實際學習成效。

研究步驟與教學實驗實施流程如下：

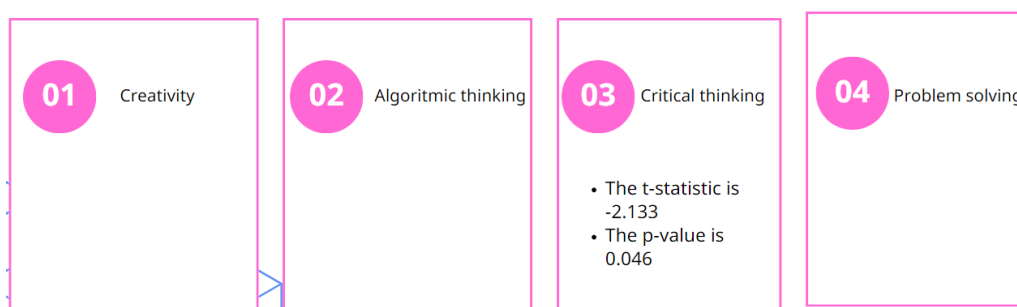
1. 學期初和學期末使用運算思維問卷，比較實驗組、對照組在運算思維量化的改變情形。
2. 參加四月份舉行的Bebras國際運算思維挑戰賽。
3. 學校教學意見期中問卷
4. 學校教學意見期末問卷
5. 撰寫實驗結論與建議
6. 檢討成果與分享經驗部分

競賽	問卷	測驗	期末報告
<u>Bebras 國際運算思維挑戰賽</u>	運算思維技能	筆試	學習歷程
<ul style="list-style-type: none"> 挑戰期間：4/22至5/03 	<ul style="list-style-type: none"> 期中問卷 期末問卷 <p>Ö. Korkmaz, R. Çakir, M.Y. Özden, A validity and reliability study of the computational thinking scales (CTS), Computers in Human Behavior, 72 (2017), pp. 558-569</p>	<ul style="list-style-type: none"> 期中考試：資料型態、判斷 期末考試：判斷、迴圈 	<ul style="list-style-type: none"> github 程式 Render 部署紀錄

5. 教學暨研究成果 Teaching and Research Outcomes

(1) 教學過程與成果

本研究按學生的作業繳交數作為上課投入情形，而將學生分為實驗組與對照組。兩組在學期初和學期末各進行一次運算思維量問卷。運算思維問卷含有 Creativity,

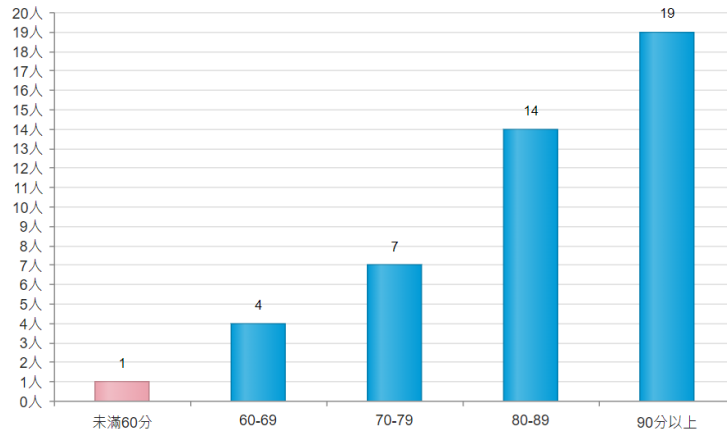


Algorithmic thinking, Critical thinking, Problem solving 等四個面向，本研究採用實驗組和對照組 t test，學期初兩組並無顯著差異。在學期末時，再次檢驗兩組的運算思維問卷，發現其中在 Critical thinking 的 p value 具有顯著差異，而在其餘的三個面向則無顯著差異。另外，計畫亦對於修課的參與情形，計算完課人數、成績及格人數、完成作業及實作活動數、回饋反思紀錄、完成 Bebras 挑戰賽人數。資料整理如下表

項目	特徵
課程整體指標	完課人數：出席週數達 10 週的人數 42 人 成績及格人數：總成績 60 分以上的人數 44 人
學生學習活動分析指標	課程評量指標： 完成作業及實作活動數 31

	<p>學習成效指標：</p> <ol style="list-style-type: none"> 1. 學生在各階段回饋反思紀錄 38 2. 完成 Bebras 挑戰賽人數 42
--	--

學期總成績分布圖如下：



(2) 教師教學反思

在近幾年的教學過程中，我觀察到多數學生在遇到學習困難時，往往會直接求助於我這位授課教師，而較少展現出自主搜尋資訊與解決問題的能力。這樣的情況引發我對自己教學方式的反思：是否是我的教學策略無法有效激發學生的自我學習動機，或是我在課堂上未能提供足夠的問題解決引導？我意識到，現代學生的學習方式似乎趨於被動，他們習慣於接受知識，而非主動探索和挖掘知識。在教學中，我曾經專注於逐步講解程式功能，希望學生能夠跟著指導一步步地掌握知識。然而，這種教學方式似乎忽略了學生需要從中學會獨立思考和自主解決問題的重要性。許多學生呈現出跳躍式思考的特點，對於指令的邏輯順序難以清晰理解，因此當我一味地累加程式功能介紹時，他們只能被動地照本宣科地複製、貼上，卻無法真正融會貫通。我開始反思，這樣的教學方式是否在無形中抹殺了學生原有的學習熱情？如果我只是傳遞知識而不去啟發他們的思考，那麼他們只會依賴教師提供的答案，而缺乏自行探索的動機與能力。這對學生長遠的發展並不利，因為他們缺乏主動

解決問題的經驗，未來在面對複雜問題時，也許會缺乏應對策略。為了改變這種情況，我必須重新檢視自己的教學方法，將重點放在培養學生的資訊搜尋能力與問題解決技巧上。我需要提供更多引導性的問題，鼓勵學生主動探索，並在遇到困難時學會使用各種資訊工具來尋找解決方案。這不僅能幫助他們掌握知識，還能培養他們在未來工作中所需的核​​心能力。透過這樣的反思，我了解到，教師的角色不應僅僅是知識的傳遞者，更應該是引導者，幫助學生從被動學習轉變為主動探索。只有這樣，我們才能真正培養出具備獨立思考和問題解決能力的學生，讓他們在未來的學習與工作中游刃有餘。

(3) 學生學習回饋

查詢教學意見 >> 查詢期中問卷

查詢範圍: 112 查詢

課程代號	112	查詢	學生留言
學年期	1122	課程代號	950074 主權量
開課系級	12000教育學院0	科目中文	120200AI思維與程式設計
編排人數	45	授課人數	11

課程代號	112	查詢	學生留言		
1.教學媒質	1.教師備課上下課	11	0	100.00	0.00
	2.教師注意出現率	10	1	90.91	9.09
	3.教師維持課堂秩序	11	0	100.00	0.00
	4.教師於課外時間提供輔導時間	10	1	90.91	9.09
	5.教師提供回答學生問題	11	0	100.00	0.00
	6.教師鼓勵學生提問	10	1	90.91	9.09
2.教學方法	7.師生互動良好	11	0	100.00	0.00
	8.教師表達方式清楚	10	1	90.91	9.09
	9.教師的教學方式能引起學生學習興趣	10	1	90.91	9.09
	10.教師鼓勵學生表達意見	11	0	100.00	0.00
	11.教師利用教學過程鼓勵學生提高學習興趣	10	1	90.91	9.09
	12.教師運用多種平台促進學生學習	10	1	90.91	9.09
3.教材內容	13.教師編寫之教材內容符合課程計畫	11	0	100.00	0.00
	14.教師之教材內容有正確知識與資訊	10	1	90.91	9.09
	15.教師之講義提供清晰簡潔使用於課堂之教材	10	1	90.91	9.09
	16.教師之教材內容能吸引學生學習	10	1	90.91	9.09
4.評量方式	17.教師採用多元化的評量方式,例如考卷與計作業設計等成績	11	0	100.00	0.00
	18.教師採用多元之評量方式	10	1	90.91	9.09
	19.教師採用多元之評量方式	11	0	100.00	0.00
	20.教師採用多元之評量方式,例如考卷與計作業設計等成績	10	1	90.91	9.09
5.針對性科目之課程內容涵蓋全面	針對性科目之課程內容涵蓋全面	10	1	90.91	9.09
6.針對性科目之課程內容適切清楚	針對性科目之課程內容適切清楚	10	1	90.91	9.09

查詢單位: 教學資源中心 教師發展組

查詢教學意見 >> 查詢期末問卷

使用須知 | 使用手冊 |

查詢範圍: 112 查詢

查詢: 學生留言

序號	學年期	開課系級-分組	科目名稱	問卷人數	選課人數	總參考評分	敬業精神	師生關係	授課方法	教材內容	教學效果	學習心得	評量方式	班平均	系平均	院平均	全校平均
1	1122	資管碩1-00	資訊管理專題研討(二)	1	6	95.2	23.8	23.8	23.8	23.8	23.8			93.65	93.95	92.60	89.77
2	1122	教育學院0-00	AI思維與程式設計	25	45	90.2	23.3	22.3	22.4	22.2	22.2			91.30	91.30	90.53	89.77

承辦單位: 教學資源中心 教師發展組

6. 建議與省思 Recommendations and Reflections

根據近幾年教學經驗發現，多數學生在碰到學習困難時，會直接求助於授課教師，而欠缺應有的資訊搜尋與問題解決能力。這樣的現象可

能源自於學生自我學習動機不足，亦或是教學者未能提供適切的問題解決引導策略。學生的自我學習動機不足，可能與現今的教育環境和文化有關，學生習慣了被動接收知識，而非主動探索。當學生遇到困難時，第一反應往往是尋求外部協助，而非自己嘗試解決問題。這樣的學習方式導致學生無法培養出獨立思考和問題解決的能力，長此以往，對學生未來的學習和發展將產生負面影響。

此外，教學者在教學過程中若未能有效引導學生自行尋找解決問題的方法，反而會加深學生的依賴心理。若教師在課堂上只是一味地累加程式功能的介紹，絕大多數學生只能被動地照本宣科地複製、貼上，對於程式的邏輯與運作方式並未深入理解。這樣的教學方式雖然能夠在短時間內讓學生完成課堂作業或測驗，但卻無法真正提高學生的問題解決能力和程式設計能力。長期而言，這不僅抹殺了學生原有的學習熱情，也限制了他們在該領域的發展。

因此，如何培養學生的資訊搜尋與問題解決能力，成為當前教育者亟需思考的重要課題。教師應該提供更多引導性的問題，鼓勵學生主動探索，讓學生在尋找答案的過程中，培養獨立思考的能力。同時，應教導學生如何有效地使用資訊搜尋工具，並分析與整合所獲得的資訊，以解決實際問題。透過這樣的教學方式，不僅可以提高學生的學習成效，還能培養他們在未來職場中所需的核心理念。總結來說，教師在教學中應扮演引導者的角色，協助學生從被動學習轉變為主動探索，從而培養出真正具備問題解決能力的學生。

二. 參考文獻 References

孔令傑 (2022), 分享運算思維 孔令傑教如何跨域教學, 淡江時報 第 1140B 期。

王佳文 (2021), 運用合作學習於非資訊領域學生學習程式相關課程之研究 - 以醫療數據分析課程為例, 教育部教學實踐研究計畫。

白瑞強 (2020), 積木程式和 Python 整合學習成效評估, 教育部教學實踐研

究計畫。

江玥慧 (2021)，為非資訊領域學生所設計之程式設計通識課程—考量學習動機與學習支援之教學實踐，教育部教學實踐研究計畫。

何素美 (2021)，運用多元化教學策略提升技職校院非資訊科系學生學習程式語言動機與成效，教育部教學實踐研究計畫。

何素美 (2022)，運用技職校院非資訊科系學生學習程式語言之專屬教材學習筆記本提升學生學習成效之研究。

何敏煌 (2018)，運用 Kirkpatrick 模式評估積木式程式設計方法對非資訊科系學生之通識程式設計課程學習興趣與學習成效之影響，教育部教學實踐研究計畫。

吳志康 (2021)，探討非資訊科系學生運用智慧型機器人為教學輔具對學習程式設計課程之學習動機、焦慮及成效之影響-- 以觀光系程式設計課程運用 Zenbo Jr.機器人為例，教育部教學實踐研究計畫。

李政雄 (2021)，以非資訊領域學生為核心之入門程式語言的教學設計，教育部教學實踐研究計畫。

邱仁一、崔夢萍 (2019)，積木式程式設計之學習成效後設分析，教育學報，第 49 卷第 2 期，頁 71 - 95。

邱奕契 (2019)，教材和教學方法對非資訊相關科系學生程式設計學習成效的影響，教育部教學實踐研究計畫。

胡逸群 (2022)，整合遊戲化教學法及參與式教學法於計算機程式課程以增進非資訊科學領域學生之學習動機及學習成效，教育部教學實踐研究計畫。

徐聖訓 (2020)，以問題導向學習為方法，培養商管學院學生大數據分析能力之研究，教育部教學實踐研究計畫。

莊益瑞 (2021)，專題式學習在非資訊科系程式設計課程之教學策略研究—以旅遊管理系為例，教育部教學實踐研究計畫。

- 陳永輝 (2019), 結合 CPBL 學習於非資訊相關科系學生在 Arduino 模組程式設計之認知學習成效差異分析與評估-以程式設計概論課程研究為例, 教育部教學實踐研究計畫。
- 陳慶帆 (2022), 科技藝術融入大一非資訊科系程式設計課程之設計, 教育部教學實踐研究計畫。
- 曾智義 (2018), 議題式程式設計教學對社會人文學生提升科技使用能力之自我效能研究: 以通識教育為例, 教育部教學實踐研究計畫。
- 黃智賢 (2022), 即時反饋的程式設計學習模式--於教學場域實踐適合非資訊科系入門的自動程式碼(c#、VB)批改系統, 教育部教學實踐研究計畫。
- 張菟珍 (1997), 鷹架理論在成人教學實務之應用, 成人教育雙月刊, 第 40 期, 第 43-52 頁。
- 葉國良 (2019), 非資訊領域學生運算思維與程式設計教學實踐之學習診斷及成效分析, 教育部教學實踐研究計畫。
- 葉國良 (2022), 融合設計思考與運算思維以強化非資訊領域學生主題式程式設計跨域專題導向學習成效, 教育部教學實踐研究計畫。
- 鄔珮甄 (2021), 運算思維自我效能量表編製與信效度驗證, 國立臺中教育大學教育資訊與測驗統計研究所碩士論文。
- 蔡逸舟 (2022), 以物聯網之「智能生活」為題發展非資訊學生程式設計應用教學之實踐與研究, 教育部教學實踐研究計畫。
- 賴志宏 (2020), 非資訊相關科系學生在混成式與線上程式設計課程中學習成效之比較研究, 教育部教學實踐研究計畫。
- 賴志宏 (2021), 線上教學影片穿插即時測驗對學習投入與學習成效影響之研究: 以非資訊相關科系之程式設計課程為例, 教育部教學實踐研究計畫。
- 鍾沛穎 (2022), 「3C」體驗式教學模式應用於程式邏輯課程學習成效之探討—以非資訊領域背景學習者為例, 教育部教學實踐研究計畫。

- Agnese, A., Pilato, G., Vassallo G., & Gaglio S. (2009). "A semantic layer on semi-structured data sources for intuitive chatbots," In: 2009 International Conference on Complex, Intelligent and Software Intensive Systems, 760–65.
- Aho, A. A. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
- Anderson, N. D. (2016). A call for computational thinking in undergraduate psychology. *Psychology Learning & Teaching*, 15(3), 226–234.
- Anshari, M., Alas, Y., Yunus, N. M., Sabtu, N. I., & Hamid, M. S. (2016). Online learning: Trends, issues, and challenges in big data era. *Journal of E-Learning and Knowledge Society*, 12(1), 121-134.
- Argal, A., Gupta, S., Modi, A., Pandey, P., Shim, S., & Choo, C. (2018). "Intelligent travel chatbot for predictive recommendation in echo platform," IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, pp. 176-183.
- Brennan, K., Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking, *Proceedings of the AREA 2012*.
- Curzon, P., Peckham, J., Taylor, H., Settle, A. & Roberts, E. (2009). Computational thinking (CT): on weaving it in. *SIGCSE Bull.*, 41, 201-202.
- Foulger, T. S., Graziano, K. J., Schmidt-Crawford, D., & Slykhuis, D. A. (2017). Teacher educator technology competencies. *Journal of Technology and Teacher Education*, 25(4), 413–448.
- Gouws, L., Bradshaw, K., Wentworth, P. (2013, October). First year student performance in a test for computational thinking. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 271–277). ACM.
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the

state of the field. *Educational Researcher*, 42(1), 38-43.

Johnson, D. W., Johnson, R. T., Roseth, C. & Shin, T. S. (2014). "The relationship between motivation and achievement in interdependent situations," *Journal of Applied Social Psychology*, 44, 622-633.

Kirkpatrick, D.L. (1998). *The Four Levels of Evaluation*. In: Brown, S.M., Seidner, C.J. (eds) *Evaluating Corporate Training: Models and Issues*. *Evaluation in Education and Human Services*, vol 46. Springer, Dordrecht.

Kong, S. C. (2014). Developing information literacy and critical thinking skills through domain knowledge learning in digital classrooms: An experience of practicing flipped classroom strategy. *Computers & Education*, 78, 160-173.

Korkmaz, O, Çakir, R. & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS), *Computers in Human Behavior*, Volume 72, 558-569.

Kukul, V., Karatas, S. (2019). Computational thinking self-efficacy scale: Development, validity and reliability. *Informatics in Education*, 18(1), 151-164.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., et al. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 33-37.

Lu, J. J., & Fletcher, G. H. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin*, 41(1), 260-264.

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, Vol. 1, p.p. 115-117.

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2018). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*.

- Reiser, B. J. (2004). Scaffolding Complex Learning: The Mechanisms of Structuring and Problematizing Student Work. *The Journal of the Learning Sciences*, 13, 273-304.
- Russell, J., Ainley, M., & Frydenberg, E. (2005). *Schooling issues digest: Student motivation and engagement*. Canberra, Australia: Department of Education, Science and Training.
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129–141.
- Sáez-López, J. M., Sevillano-García, M. L., & Sevillano-García, E. (2019). The effect of programming on primary school students’ mathematical and scientific understanding: educational use of mbot. *Educational Technology Research and Development*, 67, 1–21.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18, 351-380.
- Shawar, B. & Atwell, E. (2005). “Using corpora in machine-learning chatbot systems,” *International Journal of Corpus Linguistics*, 10, 489-516.
- Slavin, R. E. (2013). “Cooperative learning and achievement: Theory and research,” In W. Reynolds, G. Miller & I. Weiner (Eds.), *Handbook of psychology*, 7, 2nd ed. (pp.199-212). Hoboken, NJ: Wiley.
- Sletten, S. R. (2017). Investigating flipped learning: Student self-regulated learning, perceptions, and achievement in an introductory biology course. *Journal of Science Education and Technology*, 26(3), 347-358.
- Strayer, J. F. (2012). How learning in an inverted classroom influences cooperation,

innovation and task orientation. *Learning Environment Research*, 15(2), 171-193.

Sun, Z., Xie, K., & Anderman, L. H. (2018). The role of self-regulated learning in students' success in flipped undergraduate math courses. *The Internet and Higher Education*, 36, 41-53.

Syslo, M. M., & Kwiatkowska, A. B. (2013). Informatics for all high school students: A computational thinking approach. In I. Diethelm, & R. T. Mittermeir (Eds.), *ISSEP 2013. LNCS, 7780* (pp. 43e56). Heidelberg: Springer.

Talsma, K., Schüz, B., Schwarzer, R., & Norris, K. (2018). I believe, therefore I achieve (and vice versa): A meta-analytic cross-lagged panel analysis of self-efficacy and academic performance. *Learning and Individual Differences*, 61, 136-150.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Weizenbaum, J. (1966). Eliza-A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9, 36-45.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

附件 Appendix

運算思維問卷採李克特量表將結果分為五種：從不、很少、有時、經常、總是。以下是完整的 29 題問卷：

1. 我喜歡那些對自己的大多數決定都很確定的人。
2. 我喜歡那些現實和中立的人。
3. 我相信如果有足夠的時間和努力，我可以解決大部分面臨的問題。
4. 我有一種信念，當我遇到新情況時，可以解決可能出現的問題。
5. 在解決我的問題時，我相信自己能夠在制定計劃時應用它。
6. 做夢讓我的重要項目浮出水面。
7. 在尋找問題解決方案時，我相信自己的直覺和“正確”與“錯誤”的感覺。
8. 當我遇到一個問題時，在轉向另一個主題之前，會停下來思考那個問題。
9. 我可以立即建立解決問題的等式。
10. 我認為我對數學過程有特別的興趣。
11. 我認為我更容易理解用數學符號和概念進行的說明。
12. 我相信我能輕易抓住數字之間的關係。
13. 我可以用數學表達我在日常生活中遇到的問題的解決方法。
14. 我可以將口頭表達的數學問題數字化。
15. 我喜歡與我的小組朋友一起體驗合作學習。
16. 在合作學習中，我認為因為我在小組中工作，所以會取得更成功的結果。
17. 我喜歡在合作學習中與朋友一起解決與小組項目相關的問題。
18. 在合作學習中會產生更多的想法。
19. 我擅長制定解決複雜問題的計劃。
20. 嘗試解決複雜問題很有趣。
21. 我願意學習具有挑戰性的事物。
22. 我為能夠精確思考而感到自豪。
23. 在比較手頭的選項並做出決定時，我會使用系統方法。
24. 我在心中演示問題解決方案時遇到困難。
25. 我在解決問題時，不知道應該在哪裡以及如何使用變量如 X 和 Y 。
26. 我無法按計劃逐步應用解決方法。
27. 在考慮問題的可能解決方法時，我無法產生太多選項。
28. 在合作學習環境中，我無法發展自己的想法。
29. 在合作學習中，與小組朋友一起學習讓我感到疲憊。