

# 運算思維與程式設計教育浪潮

劉晨鐘\*

數位資訊科技已經被視為人類社會活動的重要組成元素，資訊科學教育在高度數位化的環境中越來越顯得重要，因此，世界各國政府已開始展開各式各樣的資訊科學教育。其中「運算思維 (Computational Thinking)」更被視為資訊科學教育中的重要核心，運算思維並非單指程式設計，它代表的是一種思考方式，是在程式設計中的重要核心能力。Jeannette Wing 教授在 2006 年第一次提出了運算思維的概念，她所指的運算思維是在執行電腦科學的相關工作時所牽涉的解決問題、設計系統、理解人類行為的思考歷程。從這個角度來看，運算思維不僅僅是寫程式的能力，還包括處理資訊的能力，以及資訊科學家通常用來解決問題的態度。更具體地說，運算思維涉及了一些重要的思維的技能，這包括抽象和分解、遞迴思維（一種電腦科學家常用來解決問題的方式）、問題分解和轉換、錯誤預防和保護，以及啟發式的推理，這些都是解決複雜問題所需要的能力，而不僅限於軟體問題。因此，許多教育學者認為運算思維應被視為不同學科的每個人（不僅是資訊科學家）都應該學習的一套通用思考方式。

在亞洲國家，運算思維成為教育實踐、研究和政策的新興議題。在臺灣，新運算思維教育於 2019 年推出，以培育 21 世紀公民所需的學習／工作技能為主旨，新課程旨在 (1) 培養運算思維技能；(2) 建立使用資訊科技的讀寫能力來進行協作及溝通；以及 (3) 培養使用資訊科技的適當態度。課程明確概述了運算思維的組成成分且該課程必須包含在學校課程當中，主題包含原始的運算思維觀念（如陣列）以及進階的概念（如樹狀和圖表）。課程將在各所學校實施，7 到 9 年級每週至少上一小時，而 10 到 12 年級至少上兩小時。

除了臺灣，世界各國亦開始在國家級的課程改革中強調運算思維與程式設計學習。韓國於 2015 年開始改革資訊教育，並強調使用資訊科技來幫助學生增強創造問題解決能力，並制定資訊科技成為中學的必修科目，每位學生必須完成至少 34 小時的資訊課程。而小學方面，5 年級 6 年級學生必須修至少 17 小時

\* 國立中央大學資訊工程學系講座教授、科技部人文司資訊教育學門召集人

的資訊課程。新加坡近年正進行智慧國家課程 (Smart Nation Programme)，其中運算思維與程式設計為實現智慧國家的重要方案，因此從學前到成人都有一些課程幫助新加坡公民認識運算思維與程式設計，從 2020 年開始，將在每個小學 6 年級實施 10 個小時的程式設計課程。遠在北歐以創新聞名的芬蘭亦將程式設計視為與閱讀、寫作、算術等一樣的核心能力，從 2016 年起將程式設計成為中小學的必修科目，而其實施的方式並不是將資訊教育列為獨立的科目，而是強調與其他學科整合，讓學生習得應用與轉換的能力<sup>1</sup>。

運算思維與程式設計教育浪潮確實在教育系統上帶來重大改變，然而以往運算思維與程式設計教育通常是在非正規課程中實現，例如社團活動、選修課程、或是網路社群中的自主學習活動。將運算思維與程式設計教育納入必修課中應謹慎施行，避免負面效果，以下就幾點建議：

## 一、興趣培養重於技能練習

程式應用軟體開發牽涉高度熱情的投入的過程，有許多軟體的開發都是程式設計師自己主動投入，而非因為外部的動機例如可以賺多少錢或得到好成績所激發。研究發現，當學生有高度的興趣主動開發程式，此時會進入心理學所謂的心流狀態，也就是一種忘記時間流逝、渾然忘我的學習狀態。在這種狀態下他會有更多的主動探索、積極實驗的最佳化學習經驗，但是這樣的狀態必須是在高度興趣下才能形成。以往在非正規的學習社群中，由於軟體的開發任務是出自於學生自己主動投入，學生能夠有高度的興趣，也比較能夠進入心流狀態。然而，當運算思維與程式設計成為必修課程之後，若是課程流於技能的練習，學生恐怕會失去對於軟體開發的興趣，反而危害到學生對於軟體開發的熱情。因此不管在學校正規教育（指大學前教育）或是坊間補習班應將學生的興趣放在首位，教師應著重在學生對於程式設計或是計算思維的興趣培養，而非技能練習。家長更應在選擇課後輔導時，選擇能夠激發孩童興趣的課程，以免學到僅是形式上的知識而喪失終身投入的熱情。

---

<sup>1</sup> 新加坡、韓國、芬蘭之運算思維與程式設計教育推動狀況引自 Seow, P., Looi, C. K., How, M. L., Wadhwa, B., & Wu, L. K. (2019). Educational Policy and Implementation of Computational Thinking and Programming: Case Study of Singapore. In *Computational Thinking Education* (pp. 345-361). Springer, Singapore. 和 Chee Kit Looi 個人聯繫。

## 二、讓學生想像力與創造力運行

若要問成功的軟體開發的關鍵，除了邏輯思維以外，創造力與想像力更是不可或缺的關鍵。軟體的開發的創意是影響軟體成功的重要因子，許多成功的軟體都是仰賴對於社會需求的敏銳觀察，在諸多的軟體中能夠開創出與獨創並有用的想法，像硬體設備一樣透過複製一個已經存在的應用軟體在市場中的價值是非常有限的。所以運算思維與程式設計能力的培養不應受限於技能練習，而應讓學生想像力與創造力在學習的過程中能夠發揮出來，充分了解軟體的開發並不是僅是程式的撰寫，更著重在問題察覺與產出獨特想法。因此，在課程規劃上應跳脫講述式與不斷練習的教學活動，反而更應著重學生自主的專案式學習，讓學生能夠透過建造心中的想像世界來達成學習的目的，以學生想像力為核心，教學的目的在鼓勵與輔助學生透過運算思維與程式設計實現他們的想像。因此，如同 Scratch 創始人 Mitchel Resnick 所主張的學習環境應該提供低門檻、高天花板並且寬廣的疆界 (low threshold, high ceiling and wide wall)，以便讓初學者能很快、並逐漸實現複雜無所拘束的想像世界。

## 三、謹慎避免評量的負面效應

如前所言，以往的程式設計學習是在非正規的環境中學習，這樣的環境是充滿自由與自主的學習氛圍，學習的評量是學生自己對自己的肯定。然而當運算思維與程式設計教育轉變成必修課程，學習評量變成不可或缺的一部分。著重於知識與技能層面的單一評量將會大大的限制學生投入熱情，並阻礙自主式的創造與學習，評量時給予學生評分結果的分數或等第亦容易造成同儕比較而損害內在興趣。因此，評量的方式應避免以上的負面效果，評量應著重了解學生的困難並給予輔助，而非僅是為了在成績單上印上分數。評量的內容除了在知識與技能，更應著重學生是否展現創意與如何達成目標的努力過程。更重要的是在情意上面，學生是否展現投入學習的熱情與克服困難展現的韌性。這是因為資訊科學家需要的不僅是技能，更需要正向的解決問題態度。

目前運算思維與程式設計已經成為教育改革的主角之一，然而這運動仍需要更嚴謹的檢視。一些必須持續關注的議題包括：兒童應該在什麼年紀開始學習運算思維？運算思維如何與現在的課程結合？初期的運算思維教育，對學生認知發展和學生動機的影響？大專非資訊相關科系學生的運算思維如何提升？這些都需要透過理論及嚴謹的方法進行檢視。目前運算思維與程式設計教育相

當受到臺灣社會重視，科技部人文司資訊教育學門即規劃相關的重點研究項目，明定資訊科學相關科系學生之資訊科學課程（例如程式設計與人工智慧）與中小學和非資訊科學相關科系（包含文、史、哲、管理、社會科學與藝術等科系）學生之資訊科學課程的教材、教法、與學習評量工具的研發與評估為資訊教育學門的重點方向。期望透過嚴謹的學術研究，能夠提供國內實施運算思維與程式設計教育的參考。