

程式語言課程之教學模式與學習工具對初學者 學習成效與學習態度之影響

陳明溥

國立台灣師範大學資訊教育研究所

摘 要

本研究透過實驗教學探討教學模式（程序導向教學 vs. 問題導向教學）與學習工具（程序式工具 vs. 物件式工具）對初學者程式設計學習成效與學習態度的影響。實驗教學活動為期五週，共計 200 分鐘，有效樣本 132 人。學習者的學習成效依認知層次分為「學習理解」與「問題解決」，分別進行分析。學習態度則是探討學習者對實驗教學活動之學習效果滿意度、教學模式接受度、及學習工具易用度的看法。

研究發現：在程式語言學習理解成效方面，初學習者並不會因為教學模式或學習工具不同而有所差異；但是，在程式語言問題解決成效方面，教學模式與學習工具交互作用達到顯著水準；在「問題導向教學模式」搭配物件式學習工具時，初學者有最佳的問題解決表現，而在其他的教學模式與學習工具組合時，學習者的問題解決表現則無顯著差異。在學習態度方面，(1)不論教學模式或學習工具之不同，各組學習者對於教學活動的「學習效果滿意度」、「教學模式接受度」、及「學習工具易用度」都是抱持正向的態度；(2)採用「問題導向教學模式」及使用「程序式工具」的初學者對於教學模式有較高的接受度；而且，(3)初學者對於「程序式工具」的易用度有較高的評價。

關鍵字：教學模式，程式語言教學，程式設計工具

壹、前言

程式設計是資訊專業領域中極為重要的基礎技能，但是學習者通常需要經過十年以上的歷練，才能由初學者逐漸成為一位精熟的程式設計專家（Winslow, 1996）。為了幫助學習者有效發展程式設計技能，在程式語言教學上不僅要讓學習者瞭解程式語言的基本結構，更需要幫助學習者培養出有效運用程式語言解決所面對問題的技能。然而，在程式設計的過程中包含許多問題解決的內隱歷程，這些歷程不容易被初學者所察覺，但卻是解決問題的重要關鍵（Govender & Grayson, 2006; Kirsner, 1998）。因此，有效培養程式設

計問題解決能力之程式語言教學方法的選擇，是影響問題解決能力培養之目標能否達成的重要關鍵。

許多程式設計教學研究（Cox & Clark, 1994; Clark, 1992; Linn & Dalbey, 1989; Mayer, 1992; Schwartz, 1988; White, 1997）發現，有些程式設計教學並未有效促進學習者問題解決能力的提升，反而因為教學方法不當或是學習者的個別差異，讓學習者無法掌握如何有效運用程式設計技能於問題的解決上，因而造成學習成效不佳。許多程式設計教學研究（Govender & Grayson, 2006; Rist,

1995, 1996; Winslow, 1996) 明確指出, 初學者在學習程式語言時, 所遭遇到最嚴重的問題就是“不知道如何將所規劃的問題解決方案以程式表達出來”。對於初學程式設計的學習者而言, 問題解決的技巧是不容易由學習者主動察覺而習得的, 因此, 教學者應該將問題解決的歷程與技巧融入教學活動中, 讓學習者透過問題解決的歷程習得程式設計的技能, 以期能有效達成程式設計教學的目標 (Clark, 1992; Mayer, 1989, 1992; Pea, 1987)。除了問題解決技巧外, 也有許多研究者 (Bennedsen & Caspersen, 2005; Bergin & Reilly, 2005; Brusilovsky & Spring, 2004;

Collins & Fung, 2002; Shanmugasundaram, Juell, Groesbeck, & Makosky, 2006; Shanmugasundaram, Juell, Lucky & Benson, 2004) 認為, 學習者所採用的程式設計工具 (如程序導向、物件導向、圖像式工具等) 也會對初學者的程式設計學習產生影響。因此, 本研究透過程序導向教學及問題導向教學的實驗教學活動, 探討在程式設計教學過程中融入問題解決的歷程, 對於初學者程式設計學習成效與態度之影響; 並透過「程序式工具」及「物件式工具」之使用, 探討不同屬性之程式設計工具對初學者程式設計學習成效與態度的影響。

貳、文獻探討

一、程式設計之問題解決歷程

在問題解決歷程的探討上, Newell 及 Simon(1972)由資訊處理之認知觀點提出「問題解決之資訊處理模式」, 以問題的心智表徵 (mental representation) 與次目標 (sub-goals) 構成之問題空間 (problem space) 來說明問題解決的資訊處理過程。Hayes (1980) 則是提出「確認問題、擬定解決計劃、蒐集與探索資訊、實際行動、及評估結果」的問題解決階段模式。Bransford 與 Stein (1984) 也提出與階段模式相近之「IDEAL 模式」, 透過 Identify, Define, Explore, Act, 及 Look back 等五個步驟, 來描述問題解決的歷程。Schunk (2003) 則認為, 常見且有效的通用性問題解決策略包括: 假設-驗證策略 (generate-and-test strategy)、方法-結果分析 (means-ends analysis)、類比推理 (analogical reasoning)、及腦力激盪 (brainstorming) 等策略。然而, Schunk 也發現, 在學習者熟悉的知識域中, 領域特定策略 (domain-specific

strategy) 比通用性問題解決策略更為有效, 但是在問題的解答不明顯的情況下, 通用性問題解決策略則是很有用的尋求解答方法; 如 IDEAL model 這一類通用性的問題解決模式, 在解決不熟悉領域的問題時最為適用 (Andre, 1986)。

將通用性問題解決歷程運用於程式設計領域時, 其問題解決歷程大致可以分為四個步驟: (1)瞭解問題需求: 主要在釐清問題所包含的相關資訊, 並瞭解與確認程式的輸入及輸出要求; (2)擬定解題計劃: 將問題分解成幾個子問題, 並以模組化的方式規劃出解決問題的步驟; (3)撰寫程式: 撰寫程式以實行所設計的解題計劃; 及(4)測試與除錯: 進行測試, 確認解題的正確性並改正錯誤 (Deek, Kimmel, & McHugh, 1998)。學習者在程式設計的問題解決歷程中, 必須選擇及整合新舊知識與規則, 以便規劃、執行、及驗證程式設計問題的解答, 因此, 將有助於其長期記憶中程式設計相關陳述性知識、規則及認知策略的整合與精緻化, 而且問題解決歷程中

學習者也必須運用分析、綜合、評鑑等高階思考技能，因而對學習者問題解決技能的精進也將有所助益。

二、以問題解決為架構之程式設計教學

程式設計的過程就是問題解決，程式語言教學的最終目標在於培養學習者透過設計程式來解決問題（Winslow, 1996）。在問題解決活動中，學習者針對待解決的問題提出可能的解決方案與策略、並經由實際行動驗證成效。學習者可以經由此假設-驗證的科學求知歷程，有效習得複雜的問題解決技能，並避免傳統學習無法學以致用及產生惰性知識的情形（Norman & Schmidt, 1992）。程式語言教學相關研究（Gagné, 1985; Mayer & Fay, 1987; Pea & Kurland, 1984; Shneiderman & Mayer, 1979）指出，在發展程式設計能力的過程中，學習者是以其所習得的階層式知識架構為基礎，逐漸發展出程式設計的問題解決技能。雖然相關文獻（Anderson, 2000; Clark, 1992; Mayer, 1992; Schwartz, 1988）也發現，程式設計學習對於學習者通用性問題解決能力並無顯著的提升，但是程式語言學習的研究重點與問題解決仍有密不可分的關係，而且許多研究者（Salomon & Perkins, 1987; Seidman, 1988; Shaw, 1986）更建議採用問題解決策略來教導程式設計學習，以期能提升學習者程式設計之問題解決能力。

我國高中電腦課程標準明確揭示，程式語言的教學目標在於「讓學習者瞭解程式語言的基本結構，以及運用程式語言解決問題」，為了達到解決問題的目標，在教學設計上應該「以問題解決為導向，再導入需要使用的相關指令與敘述」（吳正己、何榮桂，1998; 吳正己、林凱胤，1997）。Green 及 Petre（1992）也認為，我們不應該只是將程式設計看做是學習者”內在心智表徵的轉譯呈

現”，而應該將程式設計視為”程式產出的一個漸進式及探索式的問題解決歷程”。程式設計與問題解決的關係密不可分，必須以明確的問題解決方法來教授程式設計，方能有利於學習者將所習得的問題解決技能類化到其它相關的知識領域（Pedersen & Liu, 2003; Salomon & Perkins, 1987; Seidman, 1988; Shaw, 1986）。因此，以問題解決歷程作為程式設計學習的架構，將能有效使程式設計知識及程式設計之特定問題解決方法與學習情境相結合，進而能促進程式設計之問題解決成效，培養學習者運用程式設計於問題解決之技能（Deek, Kimmel, & McHugh, 1998）。

三、程式設計工具對程式設計學習之影響

不同特性的程式語言不但會影響學習者的學習經驗，對其問題解決技巧的培養與應用也是重要的影響因素（Linn & Dalbey, 1989）。因此，Cox 與 Clark（1994）建議應該依據教學目標來選用具有適切特性的程式語言，以幫助學習者達到較佳的學習成效。現今程式語言的種類繁多，但功能及用途各異，也並非所有語言均適合初學者。就程式開發環境而言，視覺化的開發環境可以提供學習者更具象化的介面，有助於初學者掌握程式設計的架構（Funkhouser, 1993）；而以文字敘述為主的程式語言環境則提供撰寫程式、編譯、除錯的程式寫作環境，使用者必須自行撰寫程式碼及建立輸出入介面。就程式邏輯而言，程序式語言（procedural language）與物件導向語言（object-oriented language）是兩大主要類別（Robins, Rountree, & Rountree, 2003）。雖然許多研究者與程式設計專家比較推崇物件導向的程式設計理念（Detienne, 1997; Rist, 1996），但是相關研究卻發現初學者對於物件的概念普遍感到困難（Bennedsen & Caspersen, 2005; Bergin &

Reilly, 2005; Shanmugasundaram, Juell, Groesbeck, & Makosky, 2006)。就規模比較小的程式設計而言，Wiedenbeck 與 Ramalingam (1999) 也發現，初學者不論使用物件導向或程序導向的語言，其學習成效間並無差異；但是當程式設計的規模大到一定程度以上時，初學者比較能掌握程序導向語言所提供的流程控制功能，而對物件導向概念的心智模式建立上則普遍產生困難。

我國高中電腦課程標準中並未指定程式語言，但早期高中資訊教材大多是以 Quick BASIC (QB) 為教學工具。QB 為 BASIC 程式語言的一種版本，屬於程序式的程式語言，提供整合式的程式開發環境，可提供設計者一個撰寫程式、編譯、與除錯的程式發展環境。近年來，由於視窗環境的蓬勃發展，許多高中資訊教材也改採視覺化的開發環境做為程式設計教學的工具，如 Visual BASIC (VB)。VB 的執行是以物件為對象，比較偏向於物件式的程式設計方式。雖然 QB 與 VB 都是 BASIC 的語法，但 VB 具有物件式特性及視覺化的開發環境，與 QB 程序導向及文字介面的程式開發環境有許多不同。White (1997) 對 QB 與 VB 的比較研究發現，初學者不論是使用 QB 或 VB 作為學習工具，對其程式設計的學習成效並無顯著的影響，但是 White 認為，物件的觀念可以讓學習者更易於瞭解資料及控制結構的內涵。Matrin (1999) 的研究則發現，VB 的方便性比較獲得學習者的青睞，但是由學習者的 VB 程式作品發現，學習者偏重於學習物件的表面性知識，反而忽略了程式設計的概念。由於「程序式程式設計工具」及「物件式程式設計工具」之效益並無定論，因此本研究透過「程序式程式設計工具」及「物件式程式設計工具」之使用，探討程式設計學習具對初學者在程式設計學習成效與態度的影響。

四、先備知識對程式設計學習之影響

資訊科學相關教學研究發現，先備知識是最能預測電腦態度與資訊技能學習成效的指標，並認為先備知識是有效習得該領域知識之必備因素，或者至少是一項促進學習的重要因素 (Aman, 1992; Clarke, 1992; Carlson & Wright, 1993)。先備知識比較豐富的學習者，在學習該領域的新知識上比較快速且有效。Taylor 與 Mounfield (1994) 明確指出，不論是物理、數學、或是電腦等專業領域，該特定領域之先備知識是專家與初學者之間最重大的差異。初學者與專家在一般性的策略與短期記憶容量大小上並無顯著差異，但二者最大的不同點就在於所擁有該特定領域知識之數量多寡、品質優劣、及如何運用。高先備知識讓專家能迅速找到許多解題的做法，並且能有效地應用這些解題的知識。Waern (1990) 也認為，學習者在心智模型的建構上必須仰賴該領域的先備知識。先備知識不足的初學者是採用由下而上的問題解決策略 (bottom-up approach)，由底層逐步拼湊出該領域知識架構的輪廓；相反地，經驗較豐富的學習者則是採用由上而下的問題解決策略 (top-down approach)，以其已知的先備知識為架構，來建構該領域知識的心智模型。Mayer 與 Sims (1994) 則認為，當學習者在執行認知任務時，具有適切的特定領域先備知識將可以彌補其能力上的不足。Weinert (1989) 的研究也發現，特定領域先備知識比學習者的智力商數更能解釋其學習的成效，而且透過適切的教學設計可以有效減低學習者先備知識差異所造成的影響。

由於學習者的先備知識會影響本研究在教學模式與學習工具對程式設計初學者的學習成效與學習態度的探討，因此，本研究在統計分析上透過共變數分析方式排除先備知

識的影響，以期能更精確檢視教學模式與學習工具對初學者程式設計學習之影響。

參、研究方法

本研究以「程式語言之控制結構」為知識域，透過網路化學習方式進行實驗教學活動。程式設計教學模式採用「程序導向教學」及「問題導向教學」，而學習工具則包括「程序式程式設計工具」及「物件式程式設計工具」。

一、研究對象

本研究以程式設計初學者為對象，考量研究對象智能發展成熟度須能適合抽象且複雜度高的程式設計學習活動，因此選定初學程式設計之高中學習者為對象。研究樣本為高中二年級修習「程式語言」課程的四個班級學生，共計 153 位學習者。在教學模式分組上，採用隨機分派方式將學習者分派為「程序導向教學組」或「問題導向教學組」，透過網路學習方式進行實驗教學活動。學習工具之分組則配合原課程之教學方式，以班級為單位隨機分派為「程序式工具組」或「物件式工具組」。排除未全程參與先備知識測驗及成就測驗表現在二個標準差以外之學習者後，有效樣本為 132 人。

本研究「程式語言之控制結構」實驗教學活動之內容，乃是配合學習者「程式語言」之課程進度於該課程第二個月開始實施，為期五週，共計 200 分鐘。四個參與實驗教學之班級，在學期初即以班級為單位，隨機分派為「程序式程式設計工具」組或「物件式程式設計工具」組。學習者在進行實驗教學活動前均已修習該課程第一個月有關程式語言基礎語法、變數、運算式、判斷敘述等課程進度，並以該班級所分派之程式設計工具進行相關實作。因此，學習者具備程式語言

基礎知識並已熟悉所使用之程式設計工具。實驗教學活動是以網路學習方式進行個別學習，學習者於該班級上課時段於電腦教室中透過網際網路登入教學系統，並由系統依據其先前隨機分派之組別提供「程式語言控制結構」之網路學習活動。

二、研究設計

本研究採用二因子設計之準實驗研究法，獨立變項為「程式設計教學模式」及「程式設計工具」；學習成效依變項包括程式設計之「學習理解」與「問題解決」；學習態度依變項則包括學習者對「學習效果滿意度」、「學習模式接受度」、及「學習工具易用度」之看法。「程式設計教學模式」依學習任務屬性與所提供之學習支持程度，區分為學習支持度較高且依循程式設計知識階層進行教學之「程序導向教學模式」及學習支持度較低且依循問題解決步驟進行學習之「問題導向教學模式」。「程式設計工具」則是依據所使用程式設計工具與開發環境屬性區分為「程序式工具」及「物件式工具」，並分別採用 Quick BASIC 及 Visual BASIC 做為本研究之程式設計工具。

學習者對於「程式語言控制結構之學習理解」乃是透過實驗教學結束後立即實施之程式設計成就測驗，評量學習者在程式碼評量 (code evaluation) 與局部程式碼產出 (code generation) 之表現；而學習者對於「程式語言控制結構之問題解決」則是透過程式設計成就測驗中之程式設計問題，評量學習者在解題規劃與完整程式碼撰寫之表現。本研究並於實驗教學結束時，透過態度問卷蒐集各

實驗組學習者對於程式設計教學之「學習效果滿意度」、「學習模式接受度」、及「學習工具易用度」之看法。

三、研究工具

本研究所使用之研究工具包括：程式設計先備知識測驗、程式語言控制結構網路學習教材、程式設計成就測驗、及學習態度問卷等。程式語言先備知識測驗之內容包含程式語言之變數、運算式、及邏輯判斷等，測驗題型則包括程式碼評估 6 題及程式碼填空 6 題，每題 1 分共 12 分。先備知識測驗採用專家效度，經學科專家審閱修正後實施，其內部一致性信度 KR_{20} 為 .76，試題平均難度為 .44，鑑別度為 .37。

程式語言控制結構網路學習教材分為「程序導向教學」及「問題導向教學」二種版本。網路學習教材之知識內容為程式語言控制結構，包括 For...Next、While...Wend 及 Do...Loop 等迴圈控制結構。「程序導向教學」乃是依控制結構主題依序提供「語法」、「概念模型」、「使用時機與範例」、及「練習」等具高度學習支持性之學習內容，幫助學習者進行學習。「問題導向教學」則是以「擲骰子遊戲」為程式設計任務，提供學習者依循定義問題、規劃解題方案、進程式碼撰寫、及驗證除錯之問題解決步驟，進行投擲二顆骰子之遊戲設計，並於解題過程中提供程式設計相關之語法、概念模型、使用時機與範例、及練習等，讓學習者依個人需求進行學習。

程式設計成就測驗乃是依據程式語言控制結構之教學目標所編製，包含控制結構之語法、語意、及應用等，測驗之題型則包括：1.程式碼評量（code evaluation）、2.局部程式碼撰寫（填空）、3.程式設計問題之解題規劃、及 4.程式設計問題之程式碼撰寫等四種。學習成效則分別針對「學習理解」與「問題解決」進行分析。「學習理解」為學習者在程式碼評量（code evaluation）與局部程式碼產出（code generation）之表現，其中程式碼評量之選擇題 10 題，局部程式碼撰寫之填空題 10 題，每題 1 分，總分為 20 分；而「問題解決」則是學習者在程式設計問題之解題規劃與完整程式碼撰寫之表現，各 10 分，總分為 20 分。測驗採用專家效度，經程式語言課程教師及學科專家審閱修正後實施，內部一致性信度 $KR_{20} = .81$ ，「學習理解」與「問題解決」之試題平均難度分別為 .62 及 .47，鑑別度則分別為 .29 及 .60。

本研究之學習態度問卷目的在蒐集各實驗組學習者對於學習效果滿意度、學習模式接受度、及學習工具易用度之看法。學習態度問卷之學習效果、教學模式、及程式設計工具易用度各 3 題，共計 9 題，採用 Likert 五點量表方式，1 代表非常不同意、2 代表不同意、3 代表部分同意、4 代表同意、5 代表非常同意。學習態度問卷各分量表之內部一致性信度（Cronbach's α ）分別為 .82、.73 及 .74，整體量表之內部一致性信度則為 .87。

肆、研究結果

本研究之資料分析乃是以程式設計先備知識測驗成績做為共變項，就教學模式與學習工具對學習成效（學習理解、問題解決）及學習態度（學習效果滿意度、學習模式接

受度、及學習工具易用度）之影響進行二因子多變項共變數分析。統計分析之顯著水準設定為 .05。本研究在進行學習成效及學習態度之多變項共變數分析前，均先透過多變項

變異數分析檢視先備知識對學習成效及學習態度之影響狀況。

一、先備知識對學習成效影響之檢視

在進行學習成效之多變項共變數分析前，先以變異數分析檢視先備知識對學習者程式設計之「學習理解」及「問題解決」表現之影響情形。學習者依據先備知識成績取前、後各約 45% 編為高先備知識組與低先備知識組，高、低先備知識組之程式設計「學習理解」及「問題解決」平均數如表 1 所示，多變項變異數分析摘要則如表 2 所示。Box's M 多變項變異數同質性考驗未達顯著水準($F = 1.128, p = .336$)，符合變異數分析之基本假設。

由表 2 變異數分析摘要可知，先備知識對「學習理解」及「問題解決」成效之影響均達到顯著水準（學習理解： $F_{(1, 111)} = 18.674, p < .001$ ；問題解決： $F_{(1, 111)} = 14.561, p < .001$ ）。由表 1 平均數可以發現，高先備知識組的學習者在「學習理解」及「問題解決」成效上均顯著高於低先備知識組學習者；顯示先備知識對學習者的程式設計學習成效具有顯著的影響。因此，本研究在資料分析上以先備知識為共變數，排除先備知識對學習者程式設計學習成效的影響，將能更精準檢視教學模式與學習工具對初學者程式設計學習成效的影響。

表 1 高低先備知識組之學習理解與問題解決平均數、標準差、及人數摘要

學習成效	先備知識	平均數	標準差	人數
學習理解成效	低先備知識組	8.98	3.72	57
	高先備知識組	11.96	3.61	56
	合 計	10.46	3.95	113
問題解決成效	低先備知識組	7.74	4.28	57
	高先備知識組	10.50	3.35	56
	合 計	9.11	4.07	113

表 2 先備知識對於程式設計學習理解與問題解決成效之變異數分析摘要

Source	Dependent variable	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識	學習理解成效	251.160	1	251.160	18.674	.000*
	問題解決成效	215.673	1	215.673	14.561	.000*
Error	學習理解成效	1492.911	111	13.450		
	問題解決成效	1644.053	111	14.811		

* $p < .05$, R square 分別為.144 及.116

二、程式設計學習成效之共變數分析

本研究學習成效分析採用二因子多變項共變數分析，以學習者程式語言先備知識為

共變數，探討程式設計之教學模式（程序導向教學 vs. 問題導向教學）及程式設計學習工具（程序式工具 vs. 物件式工具）對學習者在程式設計之「學習理解」及「問題解決」

表現之影響。在進行學習成效之多變項共變數分析前，先分別以「組內回歸係數同質性考驗」與「變異數同質性考驗」，檢視是否符合共變數分析之基本假設。組內回歸係數同質性考驗結果未達顯著水準（學習理解： $F_{(3, 124)} = .560$ ， $p = .645$ ，問題解決： $F_{(3, 124)} = .104$ ， $p = .378$ ），Box's M 多變項變異數同質性考驗亦未達顯著水準（ $F = .925$ ， $p = .502$ ），符合共變數分析之基本假設。以下分別就本研究程式設計學習之「學習理解」及「問題解決」的多變項共變數分析結果進行摘要與說明。

（一）程式設計學習理解成效之分析

學習者程式設計之學習理解成效平均數（調整後）如表 3 所示，不論是在程序導向或是問題導向之教學模式下，「程序式工具組」在學習理解之平均成效上略高於「物件式工具組」（12.79 vs. 12.06, 12.36 vs. 12.13），但是標準差也略高於「物件式工具組」（4.06 vs. 3.43, 4.26 vs. 3.80）。換言之，使用「程序式工具」之學習者在學習理解上略高於「物件式工具」，但學習者間也有較大的差異。各實驗組平均數之差異是否達到統計上之顯著水準，則由後續統計分析進一步探討。

表 3 程式設計學習理解成效之平均數、標準差、及人數摘要

教學模式	學習工具	平均數	標準差	人數
程序導向教學	程序式工具	12.79	4.06	33
	物件式工具	12.06	3.43	34
	合 計	12.42	3.74	67
問題導向教學	程序式工具	12.36	4.26	33
	物件式工具	12.13	3.80	32
	合 計	12.25	4.01	65
總 和	程序式工具	12.58	4.13	66
	物件式工具	12.09	3.59	66
	合 計	12.33	3.86	132

程式設計學習理解成效之共變數分析摘要如表 4 所示，教學模式與學習工具之交互作用與主效果均未達到顯著水準（交互作用： $F_{(1, 127)} = .111$ ， $p = .741$ ；教學模式： $F_{(1, 127)} = .098$ ， $p = .755$ ；學習工具： $F_{(1, 127)} = .938$ ， $p = .335$ ）。此結果顯示，在排除先備知識差異的情況下，學習者之程式設計學習理解成效並不會因為所採用之教學模式或學習工具不同而有不一樣的表現。由此項結果與程式設計學習理解之測驗難度（.62）可以推論，程式設計控制結構之宣告性知識內涵

對各實驗組之學習者而言並不難，各實驗組學習者均能達到適切的學習理解，並不會因為不同的教學模式或學習工具而有不同的表現。

（二）程式設計問題解決成效之分析

學習者程式設計問題解決之調整後平均數如表 5 所示，「問題導向教學組」比「程序導向教學組」有較佳的程式設計問題解決成效（10.18 vs. 8.37）；而且，在「問題導向教學模式」下，使用「物件式工具」比「程序

式工具」有較佳的問題解決成效（11.38 vs. 9.02）。各實驗組平均數之差異是否達到統計之顯著水準，則由後續分析進一步探討。

程式設計問題解決成效之共變數分析摘要如表 6 所示，教學模式與學習工具之交互作用達到顯著水準， $F_{(1, 127)} = 4.209$ ， p

$= .042$ ；顯示在排除先備知識差異的情況下，學習者在不同教學模式與學習工具下進行學習，其程式設計問題解決成效會有所不同。因此，以下分別再針對教學模式與學習工具進行單純主效果分析，以詳細探討教學模式與學習工具對學習者問題解決成效之影響。

表 4 程式設計學習理解成效之共變數分析摘要

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識（共變數）	334.743	1	334.743	26.441	.000*
教學模式	1.235	1	1.235	.098	.755
學習工具	11.877	1	11.877	.938	.335
教學模式×學習工具	1.386	1	1.386	.110	.741
Error	1607.791	127	12.660		

* $p < .05$

表 5 程式設計問題解決成效之平均數、標準差、及人數摘要

教學模式	學習工具	平均數	標準差	人數
程序導向教學	程序式工具	8.58	3.42	33
	物件式工具	8.18	3.83	34
	合 計	8.37	3.61	67
問題導向教學	程序式工具	9.02	4.72	33
	物件式工具	11.38	3.85	32
	合 計	10.18	4.44	65
總 和	程序式工具	8.80	4.09	66
	物件式工具	9.73	4.13	66
	合 計	9.26	4.13	132

表 6 程式設計問題解決成效之共變數分析摘要

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識（共變數）	223.864	1	223.864	15.752	.000*
教學模式	107.719	1	107.719	7.580	.007*
學習工具	25.795	1	25.795	1.815	.180
教學模式×學習工具	59.813	1	59.813	4.209	.042*
Error	1804.881	127	14.212		

* $p < .05$

1.教學模式單純主效果分析

教學模式單純主效果分析是以學習者之先備知識為共變數，分別針對「程序式工具組」及「物件式工具組」進行問題解決成效之共變數分析，以分別探討教學模式對「程序式工具組」及「物件式工具組」學習者問題解決成效之影響。在進行共變數分析前，先透過「組內回歸係數同質性考驗」與「變異數同質性考驗」，檢視是否符合共變數分析之基本假設。

首先，就「程序式工具組」而言，組內回歸係數同質性考驗結果未達顯著水準($F_{(1, 62)} = 2.739, p = .103$)，Levene's 變異數同質性考驗亦未達顯著水準($F_{(1, 64)} = 2.525, p = .117$)，符合共變數分析之基本假設；接著，進行「程序式工具組」問題解決成效之教學模式單純主效果共變數分析。「程序式工具組」之教學模式單純主效果分析摘要如表 7 所示，教學模式單純主效果未達顯著水準， $F_{(1, 63)} = .281, p = .632$ 。此結果顯示，在排除先備知識差異的情況下，「程序式工具組」的學習者不論是使用程序導向教學(平均數=

8.58)或是問題導向教學(平均數= 9.02)，其程式設計問題解決表現並無顯著之差別；換言之，使用「程序式工具」時，教學模式不會影響學習者的問題解決表現。

就「物件式工具組」而言，組內回歸係數同質性考驗結果未達顯著水準($F_{(1, 62)} = .169, p = .682$)，Levene's 變異數同質性考驗亦未達顯著水準($F_{(1, 64)} = .025, p = .874$)，符合共變數分析之基本假設；接著，進行「物件式工具組」問題解決成效之教學模式單純主效果共變數分析。「物件式工具組」之教學模式單純主效果分析摘要如表 8 所示，教學模式主效果達顯著水準， $F_{(1, 63)} = 12.233, p = .001$ 。由此結果及表 5 平均數可知，在排除先備知識差異的情況下，使用「物件式工具」的學習者以「問題導向教學模式」進行學習時，其程式設計問題解決之表現(平均數= 11.38)顯著優於以「程序導向教學模式」進行學習(平均數= 8.18)。換言之，使用「物件式程式設計工具」時，搭配「問題導向教學模式」進行教學活動可以讓學習者有最佳的問題解決表現。

表 7 程序式工具組問題解決成效之教學模式單純主效果共變數分析摘要

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識 (共變數)	128.955	1	128.955	8.486	.005*
教學模式	3.524	1	3.524	.281	.632
Error	957.348	63	15.196		

* $p < .05$

表 8 物件式工具組問題解決成效之教學模式單純主效果共變數分析摘要

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識 (共變數)	96.356	1	96.356	7.175	.009*
教學模式	164.292	1	164.292	12.233	.001*
Error	846.085	63	13.430		

* $p < .05$

2. 程式設計學習工具之單純主效果分析

首先，就「程序導向教學組」而言，組內回歸係數同質性考驗結果未達顯著水準($F_{(1, 63)} = .595, p = .444$)，Levene's 變異數同質性考驗亦未達顯著水準($F_{(1, 65)} = .421, p = .519$)，符合共變數分析之基本假設；接著，進行「程序導向教學組」問題解決成效之程式設計學習工具單純主效果共變數分析。「程序導向教學組」之學習工具單純主效果分析摘要如表 9 所示，學習工具單純主效果未達顯著水準， $F_{(1, 64)} = .281, p = .598$ ；此結果顯示，在排除先備知識差異的情況下，「程序導向教學組」學習者不論是使用「程序式工具」(平均數= 8.58)或是「物件式工具」(平均數= 8.18)，其程式設計問題解決成效上並無顯著差異。換言之，採用程序導向教學時，學習工具並不影響學習者的問題解決表現。

就「問題導向教學組」而言，組內回歸係數同質性考驗結果未達顯著水準($F_{(1, 61)} =$

1.578, $p = .214$)，Levene's 變異數同質性考驗亦未達顯著水準($F_{(1, 63)} = 1.555, p = .217$)，符合共變數分析之基本假設；接著，進行「問題導向教學組」問題解決成效之程式設計工具單純主效果共變數分析。「問題導向教學組」之學習工具單純主效果分析摘要如表 10 所示，學習工具單純主效果達顯著水準， $F_{(1, 62)} = 4.750, p = .033$ 。由此結果及表 5 之平均數可知，在排除先備知識差異的情況下，「問題導向教學組」學習者使用「物件式工具」(平均數= 11.38)時，其問題解決成效顯著優於使用「程序式工具」(平均數= 9.02)。換言之，以「問題導向教學模式」進行教學活動時，使用「物件式程式設計工具」可以幫助學習者達到較佳的問題解決成效。因此，採用「問題導向教學模式」進行教學活動時，「物件式程式設計工具」是較佳的選擇。

表 9 程序導向教學組問題解決成效之程式設計工具單純主效果共變數分析摘要

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識 (共變數)	84.782	1	84.782	7.027	.010*
學習工具	3.387	1	3.387	.281	.598
Error	772.220	64	12.066		

* $p < .05$

表 10 問題導向教學組問題解決成效之程式設計工具單純主效果分析摘要

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識 (共變數)	150.348	1	150.348	9.126	.004*
學習工具	78.259	1	78.259	4.750	.033*
Error	1021.394	62	16.474		

* $p < .05$

3. 教學模式與學習工具之交互作用

由程式設計問題解決成效之單純主效果分析結果可以繪製如圖 1 所示之教學模式與

學習工具交互作用圖。當程式設計教學是採用「問題導向教學模式」配合「物件式程式設計工具」的情況下，學習者在程式設計問

題解決成效上會有最佳的表現；而當教學活動是以「程序導向教學模式」搭配「程序式學習工具」或「物件式學習工具」、或是以「問題導向教學模式」搭配「程序導向程式設計工具」的情況下，學習者在程式設計問題解決的成效則無顯著的差別。由此交互作用可以得知，「問題導向教學模式」搭配「物件式程式設計工具」是提升學習者程式設計問題解決表現的有效做法。

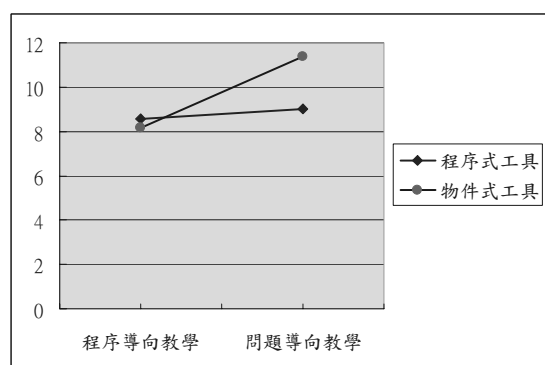


圖 1 程式設計問題解決成效之教學模式與學習工具交互作用圖

(三) 學習成效分析結果之彙整

本研究學習成效分析結果彙整如表 11 所示，在學習理解成效方面，不論初學者是採用「程序導向教學模式」或是「問題導向教學模式」，其程式設計學習理解成效相當；而且，不論學習者是使用「程序式工具」或是「物件式工具」，其程式設計學習理解之表

現也相當。推測原因，可能是學習理解主要內涵為程式設計控制結構之相關語法與程式碼撰寫之基礎知識與應用，在知識內涵的難度上並不高（.62），不論是使用程序導向教學或問題導向教學，也不論所使用的學習工具是「程序式工具」或「物件式工具」，學習者都可以有相同程度的學習表現。

在問題解決成效方面，（1）就學習工具而言，在採用程序導向教學時，不論是使用「程序式工具」或是「物件式工具」，學習者的程式設計問題解決成效相當；但是，當採用問題導向教學時，使用「物件式工具」則會有較佳的問題解決成效；（2）就教學模式而言，在使用「程序式工具」時，不論是以程序導向教學或問題導向教學進程式設計學習活動，初學者在程式設計問題解決的表現相同；但是，當使用「物件式工具」時，「問題導向教學模式」則會讓初學者有較佳的問題解決成效。

整體而言，在排除先備知識差異的情況下，採用問題導向教學搭配物件式學習工具進程式設計教學，可以幫助初學者獲致最佳的程式設計問題解決表現。因此，為了達成培養學習者程式設計問題解決能力的目標，在程式語言教學方法與學習工具的選擇上，採用問題導向教學搭配物件式學習工具是最佳的選擇。

表 11 程式設計學習成效分析結果摘要

學習成效面向	組 別	結 果
學習理解成效	教學模式	程序導向教學 = 問題導向教學
	學習工具	程序式工具 = 物件式工具
問題解決成效	程序導向教學組	程序式工具 = 物件式工具
	問題導向教學組	物件式工具 > 程序式工具
	程序式工具組	程序導向教學 = 問題導向教學
	物件式工具組	問題導向教學 > 程序導向教學

三、先備知識對學習態度影響之檢視

在進行學習態度之多變項共變數分析前，先透過變異數分析檢視先備知識對學習者之「學習效果滿意度」、「學習模式接受度」及「學習工具易用度」之影響情形。高、低先備知識組之「學習效果滿意度」、「學習模式接受度」及「學習工具易用度」平均數如表 12 所示，多變項變異數分析摘要則如表 13 所示。Box's M 多變項變異數同質性考驗未達顯著水準 ($F = 1.745$, $p = .106$)，符合變異數分析之基本假設。

由表 12 平均數可以發現，高、低先備知識組的學習者對於「學習效果滿意度」、「學習模式接受度」及「學習工具易用度」等的

看法相當接近；而就各學習態度向度之平均數而言，「學習效果滿意度」略高於「學習模式接受度」(4.23 vs. 3.94)，而「學習模式接受度」又略高於「學習工具易用度」(3.94 vs. 3.79)。由表 13 變異數分析摘要可以得知，先備知識對「學習效果滿意度」、「學習模式接受度」及「學習工具易用度」等的影響並未達到顯著水準；換言之，學習者先備知識高低並不影響其對於「學習效果滿意度」、「學習模式接受度」及「學習工具易用度」等的看法。因此，本研究在學習態度分析上不論是採用多變項共變數分析或是多變項變異數分析均為適切的做法。

表 12 高低先備知識組之學習態度平均數、標準差、及人數摘要

學習成效	先備知識	平均數	標準差	人數
學習效果滿意度	低先備知識組	4.27	1.02	57
	高先備知識組	4.20	1.08	56
	合 計	4.23	1.04	113
學習模式接受度	低先備知識組	3.95	.93	57
	高先備知識組	3.94	1.25	56
	合 計	3.94	1.10	113
學習工具易用度	低先備知識組	3.75	1.06	57
	高先備知識組	3.82	.99	56
	合 計	3.79	1.02	113

表 13 先備知識對於學習態度之變異數分析摘要

Source	Dependent variable	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識	學習效果滿意度	.132	1	.132	.120	.729
	學習模式接受度	.001	1	.001	.001	.974
	學習工具易用度	.127	1	.127	.121	.729
Error	學習效果滿意度	121.976	111	1.099		
	學習模式接受度	134.906	111	1.215		
	學習工具易用度	116.776	111	1.052		

四、學習態度之分析

本研究學習態度分析採用二因子多變項共變數分析，以學習者程式語言先備知識為共變數，探討程式設計之教學模式（程序導向教學 vs. 問題導向教學）及程式設計學習工具（程序式工具 vs. 物件式工具）對學習者「學習效果滿意度」、「學習模式接受度」、及「學習工具易用度」等看法之影響。在進行學習態度之多變項共變數分析前，先分別以「組內回歸係數同質性考驗」與「變異數同質性考驗」，檢視是否符合共變數分析之基本假設。組內回歸係數同質性考驗結果未達顯著水準（學習效果滿意度： $F_{(3, 124)} = .678$ ，

$p = .567$ ，學習模式接受度： $F_{(3, 124)} = .799$ ， $p = .497$ ，學習工具易用度： $F_{(3, 124)} = 1.122$ ， $p = .343$)；Box's M 多變項變異數同質性考驗亦未達顯著水準（ $F = .760$ ， $p = .750$ ），符合變異數分析之基本假設。各實驗組在各學習態度向度之平均數、標準差、及人數如表 14 所示，教學模式與學習工具對學習者「學習效果滿意度」、「學習模式接受度」、及「學習工具易用度」影響之多變項共變數分析摘要如表 15 所示。以下依序就「學習效果滿意度」、「學習模式接受度」、及「學習工具易用度」之分析結果分別說明。

表 14 教學模式與學習工具各實驗組在各學習態度向度之平均數、標準差、及人數

學習態度面向	組別	Mean	Std. Deviation	N
學習效果滿意度	程序導向教學組	3.85	1.15	67
	問題導向教學組	4.02	.99	65
	程序式工具組	4.03	1.07	66
	物件式工具組	3.84	1.08	66
教學模式接受度	程序導向教學組	4.08	1.07	67
	問題導向教學組	4.47	.91	65
	程序式工具組	4.46	.92	66
	物件式工具組	4.08	1.07	66
學習工具易用度	程序導向教學組	3.75	1.09	67
	問題導向教學組	3.79	.96	65
	程序式工具組	4.00	1.05	66
	物件式工具組	3.53	.95	66

表 15 教學模式與學習工具各組學習者對學習效果滿意度影響之變異數分析摘要

Source	Dependent variable	Type III Sum of Squares	df	Mean Square	F	Sig.
先備知識 (共變數)	學習效果滿意度	9.337E-02	1	9.337E-02	.080	.778
	學習模式接受度	.412	1	.412	.430	.513
	學習工具易用度	5.282E-03	1	5.282E-03	.005	.942
教學模式	學習效果滿意度	.848	1	.848	.726	.396
	學習模式接受度	4.889	1	4.889	5.106	.026*
	學習工具易用度	3.256E-02	1	3.256E-02	.032	.858

(表 15 續)

學習工具	學習效果滿意度	1.164	1	1.164	.996	.320
	學習模式接受度	4.533	1	4.533	4.733	.031*
	學習工具易用度	7.111	1	7.111	7.059	.009*
教學模式×學習工具	學習效果滿意度	1.958E-02	1	1.958E-02	.017	.897
	學習模式接受度	2.311	1	2.311	2.413	.123
	學習工具易用度	2.475	1	2.475	2.457	.119
Error	學習效果滿意度	148.402	127	1.169		
	學習模式接受度	121.611	127	.958		
	學習工具易用度	127.929	127	1.007		

* $p < .05$

(一) 學習效果滿意度分析

教學模式與學習工具對學習者「學習效果滿意度」影響之共變數分析摘要如表 15 所示，教學模式與學習工具之交互作用與主效果均未達到顯著水準（交互作用： $F_{(1, 127)} = .017, p = .897$ ；教學模式： $F_{(1, 127)} = .726, p = .396$ ；學習工具： $F_{(1, 127)} = .996, p = .320$ ）。此結果顯示，不論是「問題導向教學組」或「程序導向教學組」，也不論是「程序式工具組」或「物件式工具組」，學習者對於「學習效果滿意度」的看法並無差別。換言之，本研究的網路教學活動內容設計均能適切滿足各實驗組學習者在程式設計學習的需求，因此各實驗組學習者對於「學習效果滿意度」的看法都抱持一致的正向觀點。

(二) 教學模式接受度分析

如表 15 所示，就教學模式與學習工具對學習者「教學模式接受度」之影響而言，交互作用未達到顯著水準（ $F_{(1, 127)} = 2.413, p = .123$ ），但教學模式與學習工具主效果均達到顯著水準（教學模式： $F_{(1, 127)} = 5.106, p = .026$ ；學習工具： $F_{(1, 127)} = 4.733, p = .031$ ）。由表 14 之平均數可知，「問題導向教學組」

學習者對於所採用之教學模式的接受度（平均數= 4.47）顯著高於「程序導向教學組」（平均數= 4.08）；而且，「程序式工具組」學習者對於所採用之教學模式的接受度（平均數= 4.46）也顯著高於「物件式工具組」（平均數= 4.08）。由此結果可以得知，就教學模式而言，學習者比較喜歡將程式設計知識與技巧整合於問題解決活動之問題導向教學。推測其原因，可能是因為問題導向教學透過明確的任務與問題解決步驟，可以有效地將問題解決與程式設計知識與技巧整合於學習活動之中，使學習者能夠將所習得的程式設計知識應用於各種問題解決情境，因此使初學者對其有較高的接受度。再者，就學習工具而言，可能是「程序式工具」提供較符合初學者思考邏輯之步驟性程序，因此使初學者對於學習工具的使用比較能夠掌握，進而促使學習者透過學習工具進行習得知識的運用與驗證，因此使學習者對於其所使用教學模式產生好感。

(三) 學習工具易用度分析

教學模式與學習工具對學習者「學習工具易用度」影響之共變數分析摘要如表 15 所示，教學模式與學習工具之交互作用與教學

模式主效果未達到顯著水準（交互作用： $F_{(1, 127)} = 2.457, p = .119$ ；教學模式： $F_{(1, 127)} = .032, p = .858$ ），但學習工具主效果則達到顯著水準（ $F_{(1, 127)} = 7.059, p = .009$ ）。換言之，學習者並不會因為教學模式不同，而對所使用之學習工具易用度有不同的看法；而且，使用「程序式工具」的學習者對於學習工具易用度的看法（平均數= 4.00）顯著高於使用「物件式工具」的學習者（平均數= 3.53）。由此結果可以推論，「程序式工具」依據控制流程提供較明確且具邏輯性的程式設計程序，使初學者比較容易掌握程式設計的步驟，因此學習者認為「程序式工具」有較高的易用度；相反地，雖然「物件式工具」提供學習者較為具象化的介面，可能有助於學習者掌握程式設計的整體架構，然而，對初學者而言過於有彈性的步驟與較複雜的物件概念，可能使學習者比較不容易掌握，因而降低其對「物件式工具」易用度的看法。

（四）學習態度分析結果之彙整

學習態度分析結果彙整如表 16 所示，(1) 學習者對於本研究實驗教學之「學習效果滿意度」、「教學模式接受度」、及「學習工具易用度」等都抱持正向的肯定觀點；(2)就「學習效果滿意度」而言，由於本研究實驗教學內容與相關安排均能滿足初學者程式設計學習之需求，因此，學習者對「學習效果滿意度」都抱持一致的正向看法，不會因為教學模式或學習工具不同而不一樣；(3)就「教學模式接受度」而言，初學者比較青睞能將問題解決與程式設計知識與技巧整合於學習活動之「問題導向教學模式」；而且，「程序式工具」提供較符合初學者思考邏輯之步驟性程序，會使學習者對教學模式產生正向的看法；及(4)初學者認為，提供明確且具邏輯性程式設計程序，幫助學習者比較容易掌握程式設計步驟之「程序式工具」，比「物件式工具」有較佳的易用度。

表 16 學習態度分析結果摘要

學習態度面向	獨立變項	結 果
學習效果滿意度	教學模式	問題導向教學組 = 程序導向教學組
	學習工具	程序式工具組 = 物件式工具組
教學模式接受度	教學模式	問題導向教學組 > 程序導向教學組
	學習工具	程序式工具組 > 物件式工具組
學習工具易用度	教學模式	程序導向教學組 = 問題導向教學組
	學習工具	程序式工具組 > 物件式工具組

伍、結論與建議

程式語言課程的教學目標，不僅是要讓學習者瞭解程式語言的基本結構，更希望學習者能夠運用程式語言解決問題。由本研究學習成效之探討發現，運用問題導向教學將問題解決的歷程與技巧融入程式設計教學活

動中，可以讓學習者透過問題解決的歷程有效習得程式設計的問題解決技能。如同 Pedersen 及 Liu (2003) 的研究發現，在程式設計的問題解決過程中，學習者除了要釐清問題目標並計劃解題方案外，還必須運用所

習得的程式設計觀念及適合的指令，才能完成問題解決之程式設計任務。問題導向教學提供學習者有效的問題解決模型，使其能更容易將所習得的程式設計知識與解題技巧應用於問題情境之中，因而可以達到較佳的問題解決成效。再者，由本研究學習態度之探討也發現，學習者對於問題導向教學的接受度顯著高於程序導向教學，顯示問題導向教學不但是一種可以有效培養初學者對程式設計之學習理解與問題解決能力的教學模式，也是一種能獲得初學者青睞的學習方式。因此，就高中程式語言課程的教學目標而言，問題導向教學是達成「讓學習者瞭解程式語言的基本結構，並運用程式語言解決問題」目標的有效方法。

其次，就理想的教學設計而言，在程式設計問題解決過程中如果能提供適切的程式設計工具，幫助學習者將問題視覺化或物件化，將能促進學習者的思考與解題（Funkhouser, 1993）。由本研究的結果也證實，「物件式程式設計工具」搭配「問題導向教學模式」是提升初學者問題解決學習成效的最佳教學組合。使用「物件式程式設計工具」於問題導向教學中，也符合程式設計專家們推崇物件導向程式設計的理念（Detienne, 1997; Rist, 1996）。但是，由本研究學習工具接受度之分析也發現，初學者偏好「程序式工具」，對於「物件式程式設計工具」的接受度則較低。此結果正如同

Wiedenbeck 與 Ramalingam（1999）的發現，「物件式程式設計工具」由於較複雜的物件屬性、軟體介面、以及不若「程序式工具」那般符合初學者直覺的程式邏輯等，因而不受初學者的青睞。因此，在顧及初學者感受與維持適切學習興趣的教學考量下，程式設計工具之選用可以有二種選項：一是選擇比較符合初學者的「程序式工具」，在學習者較為熟悉程式設計技巧後再使用「物件式程式設計工具」，以幫助進階學習者能更有效處理較複雜之問題解決情境；二是使用複雜度較高之「物件式程式設計工具」，但在學習活動初期必須提供適切的工具訓練活動以幫助初學者熟練「物件式工具」之使用，使初學者在學習活動中能有效運用「物件式工具」進行程式設計任務。

基於上述分析與探討，研究者建議後續研究可以進一步探討”直接使用「物件式工具」”與”先使用「程序式工具」，再使用「物件式工具」”對學習者學習成效與態度之影響。再者，由於培養學習者問題解決能力是程式語言教學的重點目標，但現今國內尚缺乏高中程式語言教學現況與成效之全面性資訊，建議資訊教育主關機關及程式語言教學研究者可針對高中程式語言教學進行全面性調查，以期能系統性呈現高中程式語言教學在培養學習者問題解決能力之教學現況與挑戰，做為持續改進教學方法與學習工具之參考。

參考文獻

- 吳正己、何榮桂 (1998)。高級中學新訂電腦課程的內涵與特色。《科學教育月刊》，208，26-32。
- 吳正己，林凱胤 (1997)。問題解決導向的程式語言教學。《資訊教育雜誌創刊十年特刊》，75-83。
- Aman, J. (1992). Gender and attitude toward computers. In C. D. Martin & E. Murchie-Beyma (Eds.), *In search of gender free paradigms for computer education*, (pp. 33-46). Eugene, OR: International Society of Technology Education.
- Anderson, J. R. (2000). *Cognitive psychology and its implications* (5th ed.). New York: Worth Publishing.
- Andre, T. (1986). Problem and education. In G. D. Phye & T. Andre (Eds.), *Cognitive classroom learning: Understanding, thinking, and problem solving* (pp. 169-204). Orlando: Academic Press.
- Bennedsen, J. & Caspersen, M. E. (2005). Revealing the Programming Process. *Proceedings of 36th SIGCSE Technical Symposium on Computer Science Education (SIGCSE2005)*, 186-190.
- Bergin, S. & Reilly, R. (2005). Programming: Factors that Influence Success. *Proceedings of 36th SIGCSE Technical Symposium on Computer Science Education (SIGCSE 05)*, 411-415.
- Bransford, J. D. & Stein, B. S. (1984). *The IDEAL problem solver: A guide for improving thinking, learning, and creativity*. New York: Freeman.
- Brusilovsky, P. & Spring, M. (2004). Adaptive, engaging, and explanatory visualization in a C programming course. *Proceedings of the 2004 World Conference on Educational Media, Hypermedia, and Telecommunications*, 1264-1271.
- Carlson, R. & Wright, D. (1993). Computer anxiety and communication apprehension: Relationship and introductory college course effects. *Journal of Educational Computing Research*, 9(3), 329-338.
- Clark, R. E. (1992). Facilitating domain-general problem solving: Computers, cognitive processes and instruction. In E. D. Corte, M. C. Lin, H. Mandl, and L. Verschaffel (Eds.), *Learning environment & problem solving* (pp. 265-285). New York: Springer-Verlag.
- Collins, T. D. & Fung, P. (2002). A visual programming approach for teaching cognitive modeling. *Computers and Education*, 39, 1-18.
- Cox, K. R. & Clark, D. (1994). Computing models that empower students. *Computer Education*, 24(4), 277-284.
- Deek, F. P., Kimmel, H. & McHugh, J. A. (1998). Pedagogical changes in the delivery of the first-course in computer science: Problem solving, then programming. *Journal of Engineering Education*, 87, 313-320.
- Detienne, F. (1990). Expert programming knowledge: A schema based approach. In J.M. Hoc, T. R. G. Green, R. Samurcay & D. J. Gillmore (Eds.), *Psychology of programming* (pp. 205-222). London: Academic Press.
- Funkhouser, C. P. (1993). The influence of problem-solving software on student attitudes about Mathematics. *Journal of Research on Computing in Education*, 25(3), 339-346.
- Gagné, R. (1985). *The conditions of learning* (4th ed.). NY: Holt, Rinehart and Winston.
- Govender, I. & Grayson, D. (2006). Learning to program and learning to teach programming: A closer look. *Proceedings of the ED-MEDIA 2006-World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 1687-1693.
- Green, T. R. G. & Petre, M. (1992). When visual programs are harder to read than textual programs. In G. C. van der Veer, M. J. Tauber, S. Bagnarola and M. Antavolits (Eds.) *Human-computer interaction: Tasks and organization*. CUD: Rome.
- Hayes, J. R. (1980). *The complete problem solver*. Philadelphia: Franklin Institute Press.
- Kirsner, K. (1998). *Implicit and explicit mental processes*. Mahwah, NJ: LEA.
- Linn, M. C. & Dalbey, J. (1989). Cognitive consequences of programming instruction. In E. Soloway & J.C. Spohrer (Eds.), *Studying the novice programmer* (pp. 57-81). Hillsdale, NJ: Lawrence Erlbaum.
- Matrin, J. L. (1999). Teaching with Visual BASIC.

- Journal of Computer Science Education*, 13(1), 12-15.
- Mayer, R. E. (1989). The psychology of how novices learn computer programming. In E. Soloway & J.C. Spohrer (Eds.), *Studying the novice programmer* (pp. 129-159). Hillsdale, NJ: Lawrence Erlbaum.
- Mayer, R. E. (1992). Teaching for transfer of problem-solving skills to computer programming. In E. D. Corte, M. C. Linn, H. Mandl & L. Verschaffel (Eds.), *Learning environment and problem solving* (pp. 193-206). NY: Springer-Verlag.
- Mayer, R. E. & Fay, A. L. (1987). A chain of cognitive changes with learning to program in Logo. *Journal of Educational Psychology*, 79, 269-279.
- Mayer, R. R. & Sims, V. K. (1994). For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning. *Journal of Educational Psychology*, 86(3), 389-401.
- Norman, G. R. & Schmidt, H. G. (1992). The psychological basic of problem-based learning: a review of the evidence. *Acad Med*, 67(9), 557-565.
- Newell, A. & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.
- Pea, R. D. (1987). Programming and problem-solving: Children's experiences with Logo. In T. O'Shea & E. Scanlon (Eds.), *Educational computing (An Open University Reader)*. London: John Wiley & Sons.
- Pea, R. D. & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology*, 2(2), 137-167.
- Pedersen, S. & Lin, M. (2003). Teachers' beliefs about issues in the implementation of a student-centered learning environment. *Educational Technology, Research and Development*, 51(2), 57-76.
- Rist, R. S. (1995). Program structure and design. *Cognitive Science*, 19, 507-562.
- Rist, R. S. (1996). Teaching Eiffel as a first language. *Journal of Object-Oriented Programming*, 9, 30-41.
- Robins, A., Rountree, J. & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.
- Salomon, G. & Perkins, D. N. (1987). Transfer of cognitive skills from programming: When and how? *Journal of Educational Computing Research*, 3, 149-170.
- Schunk, D. H. (2003). *Learning theories: An educational perspective* (4th ed.). New Jersey: Prentice-Hall.
- Schwartz, S. (1988). *Empirical studies of a "Metacourse" to enhance the learning of BASIC*. (ERIC Document Reproduction Service No. ED 305926)
- Seidman, R. H. (1988). New directions in educational computing research. In R. E. Mayer (Ed). *Teaching and learning computer programming: Multiple research perspectives* (pp. 299-308). Hillsdale, NJ: Lawrence Erlbaum.
- Shanmugasundaram, V., Juell, P., Groesbeck, G. & Makosky, M. (2006). Evaluation of Alice World as an introductory programming language. *Proceedings of the ED-MEDIA 2006-World Conference on Educational Multimedia, Hypermedia & Telecommunications*, 1976-1982.
- Shanmugasundaram, V., Juell, P., Lucky, N., & Benson, J. (2004). Development of ViewJ: A visualization builder for object oriented programming development environment. *Proceedings of the 2004 World Conference on Educational Media, Hypermedia, and Telecommunications*, 1619-1626.
- Shaw, D. G. (1986). Effects of learning to program a computer in BASIC or Logo on problem-solving abilities. *AEDS Journal*, 19, 176-189.
- Shneiderman, B. & Mayer, R. E. (1979). Syntactic/semantic interactions in programmer behavior. A model and experimental results. *International Journal of Computer and Information Sciences*, 8, 219-238.
- Taylor, H. G. & Mounfield, L. C. (1994). Exploration of the relationship between prior computing experience and gender on success in college computer science. *Journal of Educational Computing Research*, 11(4), 291-306.
- Waern, Y. (1990). On the dynamics of mental models. In D. Ackermannn & M. J. Tauber (Eds.), *Mental models and human-computer interaction: 1. Human factors in information technology No. 3*. Amsterdam: Elsevier.
- Weinert, F. (1989). The impact of schooling on cognitive development: One hypothetical assumption, some empirical results and many theoretical speculations. *EARLIN News*, 8, 3-6.
- White, J. A. (1997). Teaching adult novices to program

with Visual Basic. *Journal of Computer Science Education*, 12(2), 15-19.

Wiedenbeck, S. & Ramalingam, V. (1999). Novice comprehension of small programs written in the procedural and object-oriented styles. *International Journal of Human-Computer Studies*, 51, 71-87.

Winslow, L. E. (1996). Programming pedagogy: A psychological overview. *SIGCSE Bulletin*, 28, 17-22.

誌謝

本研究由行政院國家科學委員會科學教育處專題研究計畫經費補助，始得以完成（計畫編號 NSC 94-2520-S-003-003），特此致謝。

作者簡介

陳明溥，國立台灣師範大學資訊教育研究所，教授

Ming-Puu Chen is a professor of Graduate Institute of Information and Computer education, National Taiwan Normal University.

E-mail: mpchen@ntnu.edu.tw

收稿日期：95.07.03

修正日期：95.10.13

接受日期：95.10.16

The Effects of Instructional Approach and Programming Tools on Novices' Learning Computer Programming

Ming-Puu Chen

Graduate Institute of Information and Computer Education

National Taiwan Normal University

Abstract

The purpose of this study was to investigate the effects of type of instructional approach and type of programming tool on novices' learning computer programming. Two types of instructional approaches, the procedural-learning instructional approach and the problem-solving instructional approach, and two types of programming tools, the procedural-programming tool and the object-oriented tool, were employed in this study. There were 153 senior high students who participated in the 5-week web-based programming learning activity. The valid sample size of this study was 132.

The performance analysis showed that (a) learners, regardless of instructional approach and programming tool, performed equally well on the comprehension test, and (b) as for problem-solving performance, the problem-solving and object-tool group outperformed the other groups. The analysis of attitudes revealed that (a) learners tended to think the received instruction was effective and they had a high degree of acceptance of it; they also thought the employed programming tool was easy to use; (b) the problem-solving approach group and the procedural-tool group showed a higher degree of acceptance of the received instruction than the other groups; and finally, (c) novices showed a greater preference for the procedural-tool than the object-tool with regard to ease-of-use.

Keywords: instructional strategy, programming language, programming tool

