國立臺灣師範大學資訊教育研究所

碩士論文

指導教授：林育慈 博士

視覺化模擬輔助類神經網路教學之研究

The Effects of Visualization and Simulation on Learning of Neural Networks

研究生：蔡政宏 撰

中華民國111年06月

**視覺化模擬輔助類神經網路教學之研究**

蔡政宏

摘要

各領域對程式設計能力之需求日益增加，然而非資訊背景的學生往往缺乏足夠的程式設計能力。傳統的程式設計課程通常著重在語法和語意的認識，較缺少培養學生在理解程式問題後的實作能力，導致學生的程式能力難以因應職場需求。因此本研究針對理學院背景的學生提出運算思維導向的程式設計教學方法，透過科學運算，讓學生經歷以建模為主的運算思維。為檢驗研究所提之教學方法的有效性，研究將針對此種教學對學生程式理解能力、程式實作能力以及運算思維進行探討。研究結果發現：本研究之學習者透過基於科學運算之運算思維導向程式設計教學後，建模品質較好的學生在程式設計學習表現上較佳，建模品質較好的學生在科學程式設計專題上的表現也較好，也發現他們對程式設計建模程序中觀察與除錯的步驟感到認同。研究相關發現可對未來運算思維教學的研究與教學者提出相關建議。

【關鍵詞】運算思維、科學運算、程式設計教學、STEM

**The Effects of Visualization and Simulation on Learning of Neural Networks**

Tsai, Cheng Hung

Abstract

Programming is playing an increasingly important role in various fields, including science. However, traditional programming instruction tends to use small-scale and general examples to explain syntax and semantic meaning of the code, which cannot foster students’ programming ability of solving real-world problems. This research was intended to develop a modelling-based instruction for scientific programming to guide students to solve programming problems based on the modelling process (phenomenon description, data modelling, algorithmic modelling, coding, and verification and debugging). A learning platform based on the proposed modelling process was also developed to assist science-major students to learn how to solve real-world scientific problems by programming. An empirical study was conducted on thirty-two science-major college students to prove the effectiveness of the modelling-based scientific programming. The experiment results show that students who engaged more in modelling had higher programming performance. The modelling-based instruction actually helps students to write programs for solving scientific problem by using both of data and algorithmic models.

**Keywords:** Computational Thinking, Scientific Computing, Programming Instruction, STEM

誌謝

蔡政宏 謹誌

中華民國一一一年六月

目錄

[摘要 I](#_Toc104300944)

[誌謝 III](#_Toc104300945)

[目錄 IV](#_Toc104300946)

[表目錄 VI](#_Toc104300947)

[圖目錄 VII](#_Toc104300948)

[第一節 研究背景與動機 1](#_Toc104300949)

[第二節 研究目的 5](#_Toc104300950)

[第三節 名詞釋義 6](#_Toc104300951)

[第一節 人工智慧 9](#_Toc104300952)

[第二節 程式設計與演算法教學 13](#_Toc104300953)

[第三節 模擬式教學 15](#_Toc104300954)

[第四節 演算法視覺化 18](#_Toc104300955)

[第一節 研究設計與架構 19](#_Toc104300956)

[第二節 研究實驗參與者 22](#_Toc104300957)

[第三節 研究程序 23](#_Toc104300958)

[第四節 研究工具 26](#_Toc104300959)

[第五節 資料蒐集與分析 34](#_Toc104300960)

[第一節 分析結果 37](#_Toc104300961)

[第二節 討論 54](#_Toc104300962)

[第一節 結論 60](#_Toc104300963)

[第二節 建議 64](#_Toc104300964)

[參考文獻 65](#_Toc104300965)

[附錄一 程式設計能力測驗 72](#_Toc104300966)

[附錄二 科學能力測驗 74](#_Toc104300967)

[附錄三 科學程式設計態度問卷 76](#_Toc104300968)

表目錄

[表 1 科學與程式設計教學時程 17](#_Toc46237678)

[表 2 建模學習程序評分等級與評分標準表 26](#_Toc46237679)

[表 3 建模高低品質組之程式設計學習描述性統計資料及ANCOVA分析 31](#_Toc46237680)

[表 4 建模高低品質組之科學學習描述性統計資料及ANCOVA分析 35](#_Toc46237681)

[表 5 學生對以科學運算來學習程式設計的想法 35](#_Toc46237682)

[表 6 建模高低品質組之專題表現描述性統計資料及T-test 36](#_Toc46237683)

[表 7 建模品質高低之期末科學程式學習態度描述性統計資料及ANCOVA分析 45](#_Toc46237684)

[表 8 學生對使用平台建模程序習程式設計的看法 46](#_Toc46237685)

[表 9 學習態度十二個面向之信度分析結果 47](#_Toc46237686)

圖目錄

[圖 1- 1本研究模擬式教學策略 4](#_Toc104380679)

[圖 2- 1模擬式教學的模型 16](#_Toc104380720)

緒論

本章共分為三節，第一節說明本研究之背景與動機以及為何使用模擬式教學輔助人工智慧概念與實作的學習；第二節提出具體的研究目的；第三節則為本研究中所提及之重要名詞的名詞釋義。

1. 研究背景與動機

當今社會中，電腦科學與人工智慧的科技應用深入我們的日常生活。人工智慧的主要研究主題(例：機器學習、深度學習)，其能夠影響我們的生活面向甚廣，像是居家機器人、交通、健康照護、教育、公共安全、職場，甚至是娛樂(Stone et al., 2016)。我國教育部也將人工智慧的概念納入108課綱中(教育部，2018)，國際上也有教育相關政策將人工智慧相關議題納入探討，像是美國 CSTA 2017 年的課綱中，針對 11 和 12 年級的學習年段，也列入人工智慧演算法的實作(CSTA, 2017)，聯合國教科文組織於2019年討論如何在教育環境中教導人工智慧(Pedro, Subosa, Rivas, & Valverde, 2019)。

　　由於人工智慧的普及，有些學者認為孩子除了具備傳統素養(例：讀寫素養、數學素養)，孩子對於人工智慧的素養也同等重要，應思考如何從小教導孩子認識人工智慧 (Kandlhofer et al., 2016; Burgsteiner et al., 2016)。從國際上到國內教育趨勢，可知思考如何教導人工智慧這項學科知識是很重要的議題。

　　然而，目前人工智慧教育相關研究，教學對象多數為大學或研究所學生(Barella, Valero, & Carrascosa, 2008; Cuéllar and Pegalajar, 2014; Fernandes, 2016; Grivokostopoulou and Hatzilygeroudis, 2014; Kochlán and Hodon, 2014; Kumar, 2004; Marković, Kostić Kovačević, Nikolić, & Nikolić, 2015; Yoon and Kim, 2015)，這對於國高中、國小階段開始有人工智慧領域的教育需求來說，沒有相關教學經驗與教學研究，在制定或發展相關教材上就會缺乏參考方針，而且這些人工智慧相關的教學研究，並沒有設計教學實驗以探討何種教學方法對學生的學習是有顯著影響的，這可能都會讓國高中、國小階段的教師缺乏理解應該使用何種方式教導學生。而人工智慧為電腦科學重要領域，也與程式設計、演算法的學習息息相關(Jacob and Warschauer, 2018)，我們若要知道此學科的教學困難，以及適當的教學方法，則應該參考過去在電腦科學或程式設計、演算法教學上有何種教學困難，了解這些困難，才得以發展合適的教材供學生學習人工智慧相關知識。

　　程式設計和演算法被認為不容易學習的原因在於，普遍學生在學習「抽象概念」時，難以將抽象概念與現實生活形成連結，進而造成學生不知道如何運用程式來撰寫問題解決的方法(Esteves, Fonseca, Morgado, & Martins, 2011; Bellstrom and Thoren, 2009)。而先前研究了解到，讓學生先理解演算法對此類知識的學習是有幫助的，製作演算法的動畫以及互動式的教學內容，也能夠讓學生更佳理解演算法的主要概念(Végh & Stoffová, 2017)。所以本研究將使用演算法視覺化與模擬式教學的方式教導人工智慧。

　　演算法視覺化(Algorithm Visualization)能夠透過圖像化的方式呈現演算法執行過程中的動態變化，比起虛擬碼或靜態圖表，更能夠幫助學生理解演算法的抽象概念(Simoňák, 2016; Tudoreanu, Wu, Hamilton-Taylor, & Kraemer, 2002)。

　　模擬式教學(Simulation Learning)被普遍使用在科學教育研究中(Jensen et al., 2002; Khoo & Koh, 1998; Luo et al., 2005)，而且依照Alessi和Trollip(2001)或De Jong 和Van Joolingen(1998)對於模擬式教學的定義，在電腦科學教育中，演算法視覺化輔助教學的研究，也屬於模擬式教學的一種，即便這些研究並不一定使用模擬式教學的詞彙描述研究中使用的教學方式(Abu-Naser, 2008; Colaso et al., 2002; Korhonen & Malmi, 2000; McNally, Naps, Furcy, Grissom, & Trefftz, 2007; Naps et al., 2003; Simoňák, 2016; Tudoreanu et al., 2002; Végh & Stoffová, 2017)，在本研究後續的敘述，為了容納更廣泛的研究領域與研究觀點，我們將用模擬式教學指稱本研究製作的教學教材。然而，不論是模擬式教學或演算法視覺化的相關研究中，都有研究得出一個共同結論說明這類型的教學方法是因為提升學生的操作和互動機會而增進學習(Colaso et al., 2002; Jensen, Self, Rhymer, Wood, & Bowe, 2002; Korhonen & Malmi, 2000; Naps et al., 2003; Saraiya, Shaffer, McCrickard, & North, 2004; Tversky, Morrison, & Betrancourt, 2002)，而且這樣的結論與一些學習理論的說法一致，互動式的學習活動能夠引起動機，進而增進學習成效(Prensky, 2002)。在Chen, Hong, Sung和Chang(2011)的研究中，也說明到模擬式教學，讓學生經歷概念學習(concept learning)、模擬操作(simulative manipulation)、概念澄清(concept clarification)，讓學生透過參數的調整學習到概念，有基礎概念後，在模擬工具上做到更詳細的概念澄清，在這過程中，學生可以經歷反思的循環。而且透過科技工具輔助學習時，要促使學習者進行「有意義的學習(meaningful learning)」，教材或教學活動需要滿足幾個精神，包含「建構」、「溝通」、「表達」、「合作」、「真實」、「反思」等(Jonassen et al., 2000)，而本研究認為模擬式教學，透過模擬工具，讓學生能夠真實地操作一個概念，進而更深層地建構知識，在課堂討論時，也更容易溝通與表達在課堂所學習到的知識。

　　而演算法視覺化的長期研究也理解出使這類教材有效的因素(Shaffer et al., 2010)，像是給予學生控制演算法的執行步驟，讓學生更深入地理解演算法(Saraiya et al., 2004)，或是將演算法的執行過程比喻成日常的生活經驗，並透過動畫呈現，更能幫助學生理解演算法的概念(Hansen, Narayanan, & Schrimpsher, 2000)。

　　依照上述人工智慧教育的趨勢，以及此學科在國高中階段的教學研究稀缺，還有模擬式教學與演算法視覺化的特色與過去的研究成果證實此教學方法能夠讓此學科的教學更有效果，本研究設計模擬式教學策略與視覺化模擬輔助教學平台，搭配人工智慧教材教導高中學生學習人工智慧領域中的類神經網路演算法。本研究的模擬式教學策略中，包含「概念理解」、「概念反思」、「概念應用」三項教學流程(如圖1)。

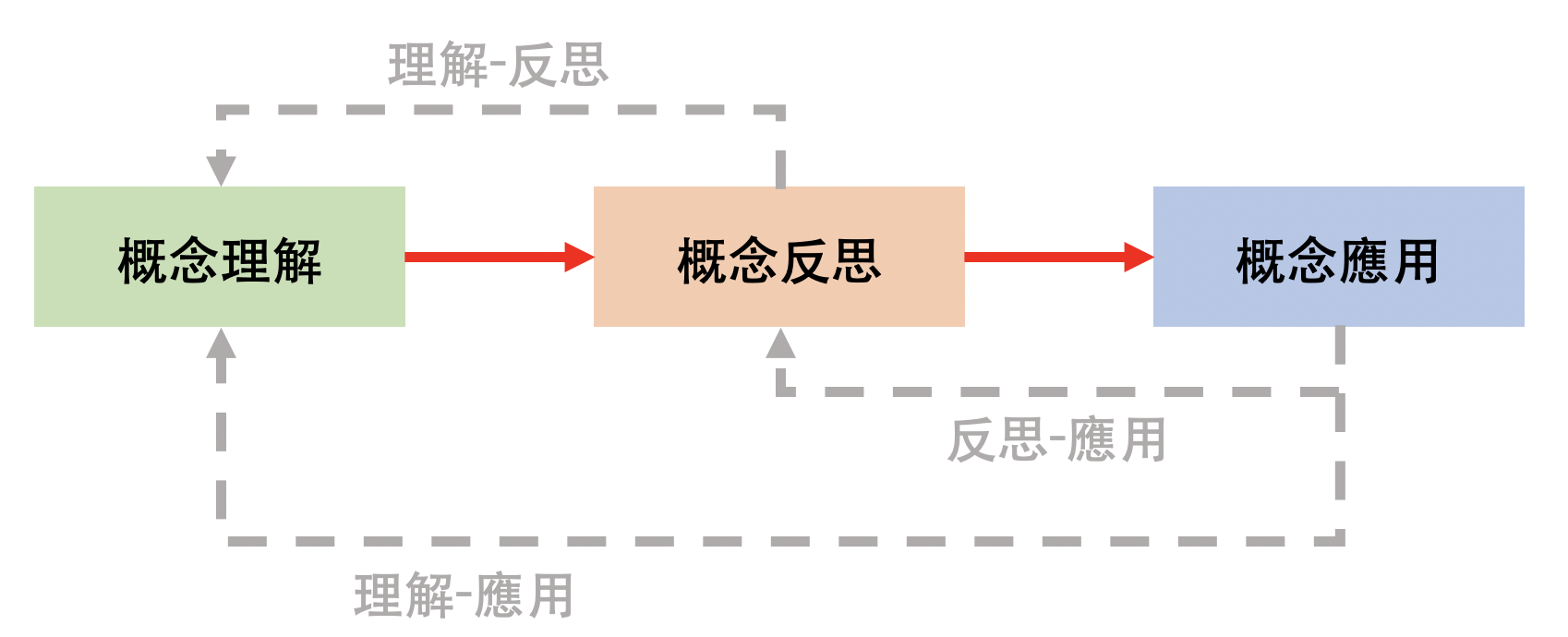


圖 1- 1本研究模擬式教學策略

　　至於教學策略中的流程如何進行，以及教學平台有哪些內容與互動的功能，本研究將在研究方法的章節中詳細說明。

1. 研究目的

本研究透過科學運算引導理學院學生學習程式設計，期望以建模與科學運算讓學生嘗試以程式設計的方式解決真實世界問題，並評估此種方式是否能夠增進學生在程式設計與科學領域的學習成效。也期許透過此種教學方式，能增進學生對於程式設計的學習態度。為此，本研究提出科學程式建模的學習策略作為課程主軸，並設計符合學生背景知識的科學運算教材，同時也架設科學程式設計建模平台作為輔助學生進行科學程式設計建模的工具。為了達到研究目的，本研究之待答問題如下：

**一、視覺化模擬輔助教學是否能提升學生對人工智慧的學習成就？**

1. 對人工智慧概念之影響。

2. 對人工智慧演算法實作之影響。

**二、視覺化模擬輔助教學是否影響學生對人工智慧的學習態度？**

1. 對電腦科學學習動機

2. 對電腦科學自我效能

3. 對資訊科學抽象概念/程序之學習感受

4. 對人工智慧學習自我評鑑

**三、學生對模擬式教學策略的課堂感受如何？**

　　1. 對「概念理解」過程的課堂感受

　　2. 對「概念反思」過程的課堂感受

　　3. 對「概念應用」過程的課堂感受

　　4. 「概念理解」、「概念反思」、「概念應用」課堂感受之比較

1. 名詞釋義

為了確保研究中所提及的各項重要名詞定義一致性，研究者將於本節作出研究內專有名詞的基本定義與解釋，以利讀者後續對論文內詞彙的解讀。

1. **理學院大學生**

本研究以STEM的教學策略嘗試將科學 (Science) 與程式設計 (Engineer) 進行科目之間的科際整合。學生須具備基本科學知識，因此本研究所提及之學生是以理學院大學生作為本研究之研究參與者，這群學生是由物理系、化學系、生科系、地科系、數學系、電機系等不同科系所組成。研究中所設計的課程題材、教學策略以及學習方法也都是以幫助理學院大學生學習程式設計並探究科學為主要考量，期望學生能夠學習科學運算並且將之使用於他們的科學專業領域學習以及未來工作上。

1. **程式設計建模程序**

學生在學習上的困難有時並非他們對概念知識的不理解，而是不知道面對問題時該如何解決，這樣的情況尤其在程式設計領域也十分常見。因此，程式設計建模程序為本研究以運算思維為主軸結合科學建模程序所發展出的程式設計學習方法，我們藉由這個程式設計建模程序帶領學生探究與學習科學運算，這項程序的內容包含：現象描述、資料建模、流程建模、程式化及觀察與除錯。當學生熟悉程式設計建模程序後，便是將該學習策略一般化與自動化，在往後他們遇到科學運算問題時，也能運用這項程式設計建模程序幫助他們在科學運算中解析並探究科學。

1. **現象描述**

在開始寫科學程式之前，學生首先要先將問題中探討的科學現象進行現象描述，這個程序的目的是引導學生從問題的本質作為思考的起點，先去找出解題所需考慮的要素、科學概念的理論與定義甚至這項定義是如何產生等。學生依序寫出或畫出他們對該主題的想法並整理思考脈絡，而這正是運算思維中將大問題分解成各種小項目的歷程，學生接著再依據這些小項目將自己的想法記錄下來，如此便能釐清自己對主題的認識，並找出寫出整個程式之前需要關注哪些面向。

1. **資料建模**

在學生完成現象描述之後，接著要找出寫科學程式所需的部分進行資料建模，這項程序則是運算思維中的抽象化，目的是讓學生能夠將原本在現象描述中所提出的各種想法轉為寫程式所需的變數，同時要詳細描述這些變數之間的關係。在平台上提供了變數、符號等方塊，讓學生能透過建立這些變數通則幫助其後續寫程式。

1. **流程建模**

流程建模則是讓學生能將原本在現象描述中提出的內容轉為撰寫科學程式所需的流程，學生能夠運用平台上流程建模中的流程方塊，整理自己寫出程式前每一項步驟的脈絡與邏輯，這樣的程序能夠讓學生在寫科學程式時有所依據，而不會因為突如其來的程式問題感到手足無措，不理解解題的順序。

1. **程式化**

待學生將現象描述、資料建模以及流程建模完成後，此時學生已能掌握自己對題目的解題策略，可以將他們對科學程式的想法以前述的建模內容加以轉為程式設計的形式，這個程序稱為程式化。

1. **觀察與除錯**

當學生將科學程式完成並執行後，則可進行觀察與除錯的程序。程式碼在執行上如果出現錯誤，他們可以根據前述自己的建模程序來觀察其中的內容是否有問題，反覆進行測試並修改。

1. **科學程式設計建模平台**

在實驗課程中，我們期望學生在開始寫程式前能夠先以運算思維找出解題的脈絡，而為了讓學生能更容易熟悉程式設計建模程序，本研究開發了科學程式設計建模平台作為輔助學生進行程式設計建模程序之用。平台上有對應程式設計建模程序的頁面，並且根據不同科學程式設計課程主題給予不同的指引。學生們在平台上建模的操作將被記錄下來供研究者進行後續分析。

1. **科學程式設計教材**

科學程式設計教材為本研究以理學院大學生學習程式設計為目標所開發的教材，研究者與科學以及程式設計之領域專家進行專家諮詢會議，共同討論教材內容並設計，採用專題導向的學習策略(Project-based learning, PBL)來設計複雜、具有挑戰性的任務，讓學生能參與解題設計、問題解決、做決策等活動，教材的內容是以他們所熟悉的科學主題進行設計，內容包含物理一維等速運動、物理拋體運動、生物DNA序列演算法、數學碎形等主題，帶領學生以程式設計建模程序來學習科學運算。

1. **科學運算學習態度**

本研究其中一項研究目的將探討此種程式設計建模程序對於理學院大學生在整體科學運算學習中的想法與感受，研究者將整體合稱為學生的科學運算學習態度，並且以李克特氏五點量表的態度問卷形式來進行調查，其內容包含學生在科學運算中的建模經驗、自我效能、學習態度以及學習興趣等四大項目。而這些又可再細分為小項目：建模經驗又可分為對現象描述、資料建模、流程建模、程式化以及觀察與除錯的建模經驗；自我效能又可分為在程式設計領域的和在科學領域的自我效能；學習態度又可分為在程式設計領域的、在科學領域的以及在科學程式設計領域的學習態度；學習興趣又可分為在程式設計領域的以及在科學領域的學習興趣。該問卷的前測與後測題目相同，而後測題目中又另外增加兩題開放式問答題，以了解學生對科學運算學習的看法，以及他們寫程式時運用哪些程式建模程序對他們釐清問題思考程式脈絡上帶來助益。

文獻探討

本章針對研究主題以及相關理論進行探討，共分為四節，第一節為人工智慧，探討人工智慧教育的趨勢以及人工智慧教學相關研究；第二節為程式設計與演算法教學，說明程式設計與演算法的教學問題，以及運算思維與其關聯；第三節為模擬式教學，探討此教學方法的特色與其應用於不同學科(例如：電腦科學、自然科學)的研究成果；第四節為演算法視覺化，探討此教學方法如何幫助學生學習演算法以及其相關研究成果。

1. 人工智慧
2. **人工智慧教育的趨勢**

現今人工智慧的主要研究主題(例：機器學習、深度學習、機器人、電腦視覺、物聯網、自然語言處理)，其能夠影響我們的生活面向甚廣，像是居家機器人、交通、健康照護、教育、公共安全、職場，甚至是娛樂(Stone et al., 2016)。人工智慧這項科技，能夠嵌入於硬體的機械上，也能夠應用於軟體的運作與開發上，人們經常使用的行動裝置，及其提供的應用與服務，皆有可能是基於人工智慧所開發的，例如：Google、Siri、Cortana、自駕車、聊天機器人(Kandlhofer, Steinbauer, Hirschmugl-Gaisch, & Huber, 2016; Burgsteiner, Kandlhofer, & Steinbauer, 2016; Estevez, Garate, Guede, & Grana, 2019)。

　　我們所處的生活環境，食衣住行育樂再也難以與人工智慧科技分離，所以我們應該培養孩子擁有足夠的能力理解這個充斥著人工智慧的社會，我國教育部也將人工智慧的概念納入108課綱中(教育部，2018)，也針對國內中小學設計人工智慧相關教材《和AI做朋友》(黃仁暐、涂益郎，2019），在美國 CSTA 2017 年的課綱中，針對 11 和 12 年級的學習年段，也列入人工智慧演算法的實作(CSTA, 2017)，聯合國教科文組織於2019年也討論人工智慧在教育領域中的挑戰與機會，探討人工智慧如何影響教育，重新思考如何培養學生擁有程式設計能力以面對現今社會(Pedro, Subosa, Rivas, & Valverde, 2019)。可見在國際上，人工智慧與教育的連結實屬重要，不論是在生活面向或教育政策面向，我們都應該重視人工智慧這項議題。

　　然而，由於人工智慧已經與我們的社會密不可分，也有學者認為孩子除了具備傳統素養(例：讀寫素養、數學素養)，孩子對於人工智慧的素養也同等重要，應思考如何從小教導孩子認識人工智慧，並且根據Russell 和 Norvig (2002)所撰寫的人工智慧教科書，訂定人工智慧素養，其中包含「自動化」、「智慧代理」、「圖論與資料結構、電腦前學基礎知識」、「排序」、「透過搜尋解決問題」、「經典規劃」、「機器學習」 (Kandlhofer et al., 2016; Burgsteiner et al., 2016)，訂定出明確的人工智慧素養，也使教學或研究更容易界定何種教學主題屬於人工智慧教育。

　　另外，由於人工智慧相關知識與電腦科學息息相關，近期電腦科學教育領域討論的運算思維，也與人工智慧的知識有所關聯(Jacob and Warschauer, 2018)，所以本研究除了回顧人工智慧教育相關研究，也同時參考了運算思維與程式教育相關研究以發展此教材，其內容會在後續的章節說明。

1. **人工智慧教育的相關研究**

目前人工智慧教育相關研究，教學對象多數為大學或研究所學生(Barella, Valero, & Carrascosa, 2008; Cuéllar and Pegalajar, 2014; Fernandes, 2016; Grivokostopoulou and Hatzilygeroudis, 2014; Kochlán and Hodon, 2014; Kumar, 2004; Marković, Kostić Kovačević, Nikolić, & Nikolić, 2015; Yoon and Kim, 2015)，這對於國高中、國小階段開始有人工智慧領域的教育需求來說，沒有相關教學經驗與教學研究，在制定或發展相關教材上就會缺乏參考方針，而且在大學階段的教學研究，很有可能學生已經具備一定程度的學科專業知識(例：Cuéllar和Pegalajar(2014)教導的學生為電腦工程科系大學三年級的學生)，與國高中、國小階段的學生有相當大的學科能力差距，以至於這些教學案例或研究，對於此學習階段的參考價值並不高。

　　而且大學階段以上的教學研究，有的教學研究僅僅介紹了課程架構、內容，或是課程進行的方式，並沒有設計教學實驗進行研究，像是Cuéllar和Pegalajar(2014)使用LEGO機器人發展教材來教導機器人程式設計、規劃演算法、審議型代理、搜尋演算法等等教學主題，有進行實際教學，但對於實際上解決什麼教學問題，論文當中沒有明確寫出，而是在硬體設備和專案的內容有很詳細的交代，而Yoon和Kim(2015)則是透過憤怒鳥遊戲發展人工智慧教材，利用遊戲式學習與專案導向式學習策略，教導學生策略決策制定、電腦視覺、資源管理等主題；有的研究則是介紹教導人工智慧課程適合使用的硬體，或是軟體環境，像是Barella, Valero和 Carrascosa(2008) 開發一個多智慧體系統(Multiagent System)，讓學生能夠在這個系統上開發自己的智慧代理人，並且加入奪旗遊戲(Capture-the-Flag)的機制，讓學生們開發的智慧代理人能夠互相競爭，而Grivokostopoulou和Hatzilygeroudis(2014)則是透過語意網路(Semantic Web)發展個人化學習的教材來教導人工智慧相關主題，類似的研究還有Marković等人(2015)發展智慧型系統讓學生進行自主學習人工智慧相關知識，Kochlán和Hodon(2014)介紹Yrobot這個開放式硬體的架構與功能，並說明此硬體能夠發展人工智慧教材。上述在大學階段的教學研究，雖然提供了人工智慧教學主題的實際案例或方法，但皆無採取分組教學的研究方法，這樣將難以檢驗何種教學設計實際影響了學生的學習成效，並且也只有少數研究確實說明人工智慧教學的疑問(例如：機器人專案對於學生習得人工智慧概念是否有效率？)且試圖使用教學數據回答這些問題(Kumar, 2004)。

　　高中階段的教學研究則有探討學生參與機器人競賽的影響(Sklar, Eguchi, & Johnson, 2002)，但其中也只有調查學生對於參與競賽的感受，並沒有對於學生的學科能力改變進行調查。

　　即便現有人工智慧素養相關研究的出現，讓幼兒階段，以及國中、國小階段有可行的人工智慧課程，但其相關研究的評估方法也僅旨在確認有達成研究自身訂定的教學目標(Kandlhofer et al., 2016; Burgsteiner et al., 2016)，同樣沒有透過教學數據驗證何種教學設計影響著學生的學習成效。

　　然而，先前人工智慧教學研究較少提及此教學主題上可能會遇到的困難(Barella, Valero, & Carrascosa, 2008; Cuéllar and Pegalajar, 2014; Fernandes, 2016; Grivokostopoulou and Hatzilygeroudis, 2014; Kochlán and Hodon, 2014; Kumar, 2004; Marković et al., 2015; Yoon and Kim, 2015)，這也可能因為人工智慧教學為現代的新興教學主題，所以較無前人整理此學科的教學問題與困境。既然人工智慧這個學科屬於電腦科學，也與程式設計、演算法的學習息息相關(Jacob and Warschauer, 2018)，所以本研究認為應該要參考相關學科的教學研究，藉此了解人工智慧教學可能會遇到的困難，知道教學上的困難，才得以運用合適的教學方法解決這些困難。

　　在第二節「程式設計與演算法教學」，我們將會透過探討程式與演算法教學上的困難，思考人工智慧教學中也可能遇到的困難，再透過第三節、第四節介紹「模擬式教學」與「演算法視覺化」的特色發展教材以克服這些教學困難，並且以教學實驗的方式檢驗這樣的教學設計對學習有實質幫助。

1. 程式設計與演算法教學
2. **運算思維**

隨著資訊科技的進步，培養學生問題解決能力、運算思維能力，程式設計與演算法成為二十一世紀重要的學習科目，在人工智慧素養被提出之前(Kandlhofer et al., 2016; Burgsteiner et al., 2016)，已有學者提出運算思維的概念以及其重要性，運算思維(Computational thinking)被描述為人類思考問題解決的一種方式，利用電腦科學的基本概念進行問題解決的思維(Wing, 2006)。如果學生擁有運算思維，當學生在面對問題時，就有機會運用電腦科學家的思維進行問題解決(Selby & Woollard, 2013)，在這日常生活充斥著科技的時代，學生擁有運算思維才有能力善用或掌握科技。Selby和 Woollard(2013)認為運算思維著重問題解決的過程，擁有抽象(Abstraction)、分解(Decomposition)、演算法設計(Algorithmic design)、評估(Evaluation)、一般化(Generalization)的思維過程。

　　透過學習程式設計與演算法，學生能夠更加貼近電腦科學家的思維方式，並且因為運算思維、程式設計與演算法、人工智慧，本身就是電腦科學領域的相關知識(Jacob and Warschauer, 2018)，所以本研究抱持著程式設計與演算法是學生認識人工智慧的基石，以及培養運算思維的途徑之一，了解過去相關教學研究，將更能發展合適的教材教導學生人工智慧，並且培養學生運算思維的能力。

1. **程式設計與演算法教學的困難**

程式設計和演算法被認為不容易學習的原因在於，普遍學生在學習「抽象概念」時，難以將抽象概念與現實生活形成連結，進而造成學生不知道如何運用程式來撰寫問題解決的方法(Esteves, Fonseca, Morgado, & Martins, 2011)。

　　學生在學習程式設計或演算法時，通常需要四種基礎知識才能熟悉程式設計的技巧(Bellstrom and Thoren, 2009)：

1. 解決問題的基礎數學知識。

2. 了解如何操作撰寫程式的環境(programming environment)的知識，例如：如何編譯與執行程式碼？如何進行程式偵錯？

3. 程式語言相關知識，例如：資料結構、函式，或是特定程式語言的語法。

4. 將整體學習到的知識轉化成程式邏輯的知識，例如：使用特定的資料結構、函式、控制流程做特定問題的解決方法。

　　然而，第四種知識對於學生而言相對困難，製作演算法的動畫以及互動式的教學內容是能夠有效解決這個困難的 (Végh & Stoffová, 2017)，因為先前研究發現動畫與互動式的教材能夠幫助學生學習程式設計與演算法，所以本研究透過模擬與演算法視覺化的方式來輔助學生學習人工智慧。

1. 模擬式教學
2. **模擬式教學的定義**

　　模擬式教學依照Alessi和Trollip(2001)的定義是將一些現象或活動呈現給學生，並且讓學生能夠與模擬環境互動，以進行學習活動。De Jong 和Van Joolingen(1998)認為，模擬式教學是一個程式，它呈現系統性的模型或過程，模擬可以廣泛地分為兩類：第一、融入概念模型的模擬；第二、融入操作模型的模擬；概念模型中，會有學科的概念或原則，像是經濟學、物理學的原理；操作模型中，會有認知或非認知的操作流程，像是雷達操作。模擬的過程中，可以改變輸入的變數達成學習上所需的輸出狀態，或是檢驗輸出狀態，模擬可以融入非線性結構的學習目標，學習目標一旦達成，學生仍然能持續和反覆調整變數(O'Neil, Wainess, & Baker, 2005)。

　　學生透過模擬，在操作、調整參數的過程中，可以檢驗自己習得的概念，在模擬工具給予回饋時產生認知衝突，或是有機會在傳統實驗或高風險實驗以外的情況，了解實驗的變數與過程，學生也能夠反覆運用視覺化模型反思自己的概念，並且不斷詢問「如果…」與「為什麼」這類的問題(Chen, Hong, Sung, & Chang, 2011; Huppert, Yaakobi, & Lazarowitz, 1998; Thomas, & Neilson, 1995)。

1. **模擬式教學之實施**

模擬式教學能夠提升學生互動與操作的機會，這些互動與操作能夠增進學生的學習成效(Chen, Hong, Sung, & Chang, 2011)。Chen, Hong, Sung和Chang(2011)將模擬式教學的模式(如圖2)，分成三個面向：概念學習(concept learning)、模擬操作(simulative manipulation)、概念澄清(concept clarification)，學生透過參數的調整學習到概念，在學生有基礎概念後，用模擬工具達到更詳細的概念澄清，在這過程中，學生可以經歷反思的循環。

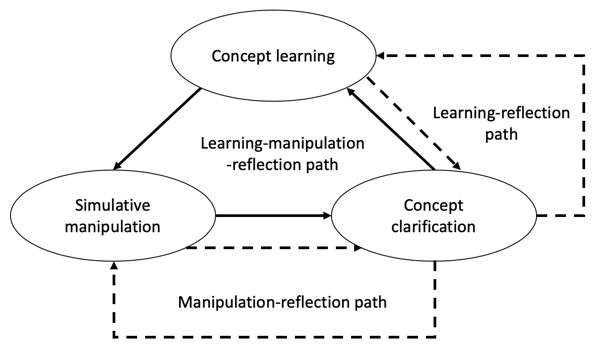


圖 2- 1模擬式教學的模型

(Chen, Hong, Sung, & Chang, 2011)

　　在教學的過程中，模擬式教學能夠幫助學生理解抽象概念，因為一些靜態圖像難以表達動態的抽象概念，像是在醫學模擬訓練中，動態的模擬更能夠使學生理解藥物對血壓造成的影響(Krishnan, Keloth, & Ubedulla, 2017)，視覺化的模擬工具也能夠幫助學生了解電腦硬體之間的運作的抽象概念，讓學生清晰地理解電腦科學領域中的抽象定義(Garay et al., 2017)，在模擬環境中，學生能夠反覆操作，更能促進精熟學習，比起講述式的課程，知識與技巧的留存會更加持久，模擬給予學生適當的訓練，根據學生自己的學習情況調整學習步調，促使學生能自主學習，讓他們在面對與模擬情境類似的情況時，更有信心完成任務(Hošková-Mayerová & Rosická, 2015)。

　　在科學教育方面，模擬環境的即時輸入與輸出，讓學生判斷系統上各種元素間的關係，這些關係是根據清楚界定的假設，與傳統實驗不同的是即時的呈現，能夠讓學生更容易得知假設與實驗結果間的關係(Mintz, 1993)，在模擬工具上，對學生要求更多探索，會讓學生處理模擬工具的過程更好，而且也會有更深層的學習效果，探索的過程也會讓學生的理解更好，學生與模擬工具互動程度也與他們轉換到真實實驗的學習成就有顯著相關(Homer & Plass, 2014)，教學內容如果牽涉到複雜的計算，但教學目的是期待學生了解一個整體性的概念，模擬工具可以協助一些計算過程，減輕學生的認知負荷，讓他們在與模擬工具互動時，更加專注於整體的概念運作(Moyer-Packenham et al., 2019)。

　　根據上述優點，因為本研究的教學主題為人工智慧中的類神經網路，認為類神經網路的運算架構屬於抽象概念，且學生需要反覆的演算類神經網路的架構進而熟悉此概念，所以使用模擬式教學，在類神經網路視覺化的模擬環境中，讓學生自主操作得到即時的輸出反饋，了解到類神經網路這個抽象概念，而模擬環境也會協助運算類神經網路中的數學概念，使學生更著重於學習類神經網路的整體概念。

1. 演算法視覺化

演算法視覺化是用圖像化的方式呈現演算法或程式動態改變的運行過程，視覺化的主要目的是幫助學生更加理解演算法(Tudoreanu et al., 2002)。在電腦科學教育中，視覺化也時常被討論是否能夠提升教學效率(Diehl, 2007; Tudoreanu et al., 2002; Shaffer et al., 2010)。

　　在過去演算法視覺化的相關研究中，有部分研究說明這類教學對於學生的學習成效沒有顯著地幫助(Gurka & Citrin, 1996; Hundhausen & Douglas, 2000; Jarc, Feldman, & Heller, 2000) 在演算法視覺化的教材中，若給學生建立自己的視覺化作品會使他們分心，而且視覺化的美觀程度還有是否使用電腦作為媒介對於教學成效沒有顯著差異(Hundhausen & Douglas, 2000)。與演算法視覺化教學系統有互動機會的學生，相較於沒有機會互動的學生而言，可能在某些類型的問題上表現良好，但有互動機會的學生需要花費較多時間在這些教材上，通盤來說仍然表現較差(Jarc et al., 2000)。有的研究則明確地指出這類教學是確實有幫助到學生學習的(Shaffer et al., 2010; Lawrence, Badre, & Stasko, 1994; Hundhausen, Douglas, & Stasko, 2002)，對於演算法視覺化的長期研究也理解出使這類教材有效的因素(Shaffer et al., 2010)，像是給予學生控制演算法的執行步驟，就能夠幫助學生拆解演算法的邏輯，進而更深入地理解演算法(Saraiya et al., 2004)，或是透過比喻的方式，將演算法的執行過程比喻成日常的生活經驗，並透過動畫呈現，能夠幫助學生理解，也讓學生能夠長期記憶演算法的概念(Hansen, Narayanan, & Schrimpsher, 2000)。

　　依照上述演算法視覺化的研究成果，本研究在設計模擬式教學輔助工具時，會讓學生能夠操作與觀察類神經網路執行過程中的每個步驟，也會使用較日常生活的例子說明類神經網路的運作原理與應用方式。

研究方法

本章節共分為六節，第一節說明本研究之設計及研究架構；第二節說明研究參與者之相關背景及能力；第三節說明本研究之實驗程序；第四節說明本研究使用之工具；第五節為視覺化模擬輔助教學設計，說明視覺化模擬平台設計與視覺化模擬教學策略；第六節說明研究蒐集的資料項目與如何進行分析、討論。

1. 研究設計與架構

本研究欲探討程式設計與科學單元之STEM跨領域教學，對學生學習程式設計以及科學的影響。因此本研究結合運算思維及科學建模程序，提出程式設計建模程序，以建模導向科學程式設計教學的方式，讓學生在以科學為主題的程式設計課程中學習Python程式語言，並探究其在程式設計學習、科學學習、專題表現以及科學運算學習態度上之成效。

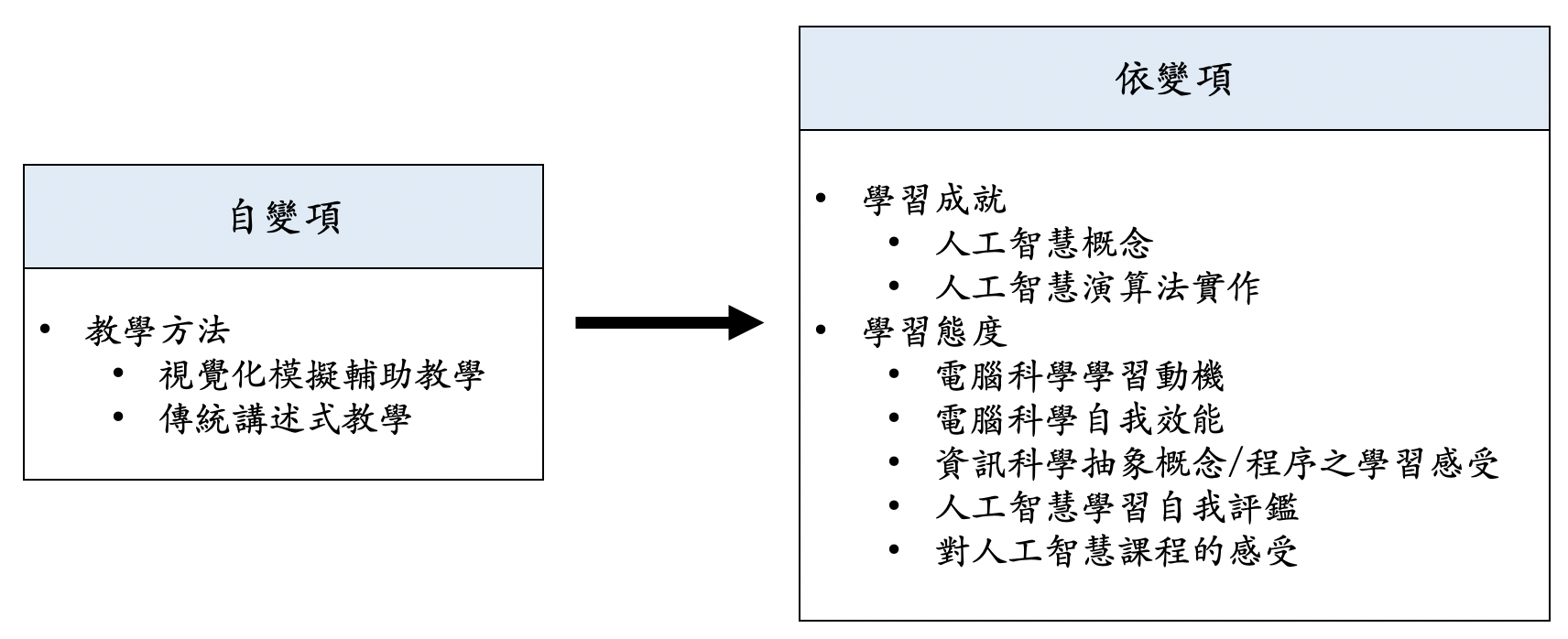


圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑2研究架構圖

**一、科學程式設計教材**

實驗開始前，研究者與專家進行多次諮詢會議，並且透過會議中的討論與建議設計出科學程式設計教材，以利後續實驗使用。科學程式設計教材是以科學概念結合程式設計概念為發想，與傳統的程式設計教學方式不同，以STEM的教學方式將程式設計結合建模與科學運算，帶領學生在學習程式設計的同時也能探究科學。透過經歷真實世界的科學情境，使學生能以他們所熟悉的科學現象加深對程式語法的理解與應用，也能從程式的操作上體會到兩者之間的連結。

在科學程式設計教材中程式設計的部分，包含基本程式語法結構(Structure)、程式執行程序(Sequence)、變數(Variable)、條件判斷(Condition)、迴圈(Loop)、串列(List)等概念，並且給予課程概念相關的練習題，讓學生能在課程結束後加深對程式概念的熟悉感；在科學程式設計教材中科學的部分，是以理學院學生熟悉的物理一維等速運動、物理拋體運動、生物DNA序列以及數學碎形等單元為主題。

**二、教學策略**

本研究的教學策略是提供給尚未熟悉程式概念的學生們一項學習程式設計的方式：程式設計建模程序。在學生開始寫程式之前，教學者會針對科學程式設計課程中所探討的科學主題，逐步引導學生在科學程式設計建模平台上進行程式設計建模程序，並期望學生能透過程式設計建模程序幫助他們學習程式設計。

學生在教學者的問題引導下開始進行現象描述，學生首先必須理解模擬程式中涵蓋的科學問題以及其背後的原理，同時也須思考要寫出模擬該科學現象的程式會使用哪些要素，學生將問題的邏輯完整地想過一遍後加以表徵，並在平台上回答，越細緻的思考將有利於學生寫出完整的程式。接著依據描述的內容建立資料模型與流程模型，在資料建模時，學生必須思考現象描述中的文字能提取出哪些重要的變數，也必須將寫出程式所需的步驟記錄在流程建模中。待模型建立完成後才進行程式化的步驟。程式設計完成之後學生須對自己寫的程式進行觀察與除錯，最後能夠敘述程式撰寫的想法並進行公開發表。如圖 2程式設計建模程序所示。



圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑3 程式設計建模程序

1. 研究實驗參與者

為驗證基於科學運算之運算思維導向教學對培養學生程式設計能力之成效，研究者將課程開發為適用於理學院大學生的內容，實驗課程參與者為大學通識課的理學院學生38名，排除未參與課程及未繳交期末專題的6名學生後，最後的32名學生為本次實驗的研究對象。

本次研究的參與者為臺北市某大學的理學院大學生，這些學生具備基本的程式設計邏輯概念與一定程度的科學知識，學生們曾經學過Scratch或Blockly等圖像式程式語言，因此對於程式設計的邏輯有初步的理解與認識，但對文字式的程式語言以及Python語法不熟悉，需要有進一步的教學來引導學生學習程式設計。

1. 研究程序

研究籌備階段，研究者先與程式設計專家以及科學專家進行晤談，並依照專家的建議設計科學程式設計課程教材以及開發科學程式設計平台的各項功能，以確保將科學概念結合程式設計學習之目的得以達成。

在正式實驗開始前，研究者在一個假日舉行了為期兩天的科學程式設計工作坊，招募理學院的學生作為工作坊的參與者進行一次實驗。依照本次工作坊的實驗成果與建議作為課程修改的參考，並且將教材內容與課程進行適切的調整。

正式實驗於工作坊結束後實施，是為期一學期的科學與程式設計通識課程。本研究包含程式設計與科學概念前測、課程內容、課堂練習、科學問題解題、程式設計與科學概念後測、科學運算學習態度問卷。首先，在課程開始前先進行課程說明，為確保學生有同等程式設計概念，教師進行共計8週的Python教學。課程於第9至第16週開始依照不同主題進行科學程式設計教學，在每次課程中，學生必須依據課程探討的科學主題，經由教師的引導在程式設計建模平台上先依序建立該科學現象的解題模型，包含現象描述、資料建模、流程建模、程式化以及觀察與除錯，透過程式設計建模的訓練，練習分析拆解科學問題、找出解題所需要素，並設計解題所需的演算法。配合學校期中考日期，實驗於第11週進行前測，測驗內容分為程式設計能力、科學能力及科學運算學習態度等測驗，以了解學生各方面程度。課程第17週時，學生可針對自己的專題內容在課堂中與老師或同儕討論。最後在第18週時，進行學生的專題發表以及實驗後測，測驗內容包含學生程式設計能力、科學能力以及科學運算學習態度等測驗。如表 1研究流程所示。

表 1 科學與程式設計教學時程

|  |  |  |
| --- | --- | --- |
| 週次 | 主題 | 課程內容 |
| 1 | 科學與運算 | 科學運算簡介 |
| 2 | 科學與運算 | 科學、模型與程式設計，練習科學建模平台 |
| 3 | 科學問題解決 | 以科學問題練習Python 3的input、print和integer |
| 4 | 科學問題解決 | 以科學問題練習Python 3的if then else condition |
| 5 | 科學問題解決 | 以科學問題練習Python 3的for loop和while loop |
| 6 | 科學問題解決 | 以科學問題練習Python 3的function和recursion |
| 7 | 停課一次 | |
| 8 | 科學問題解決 | 以科學問題練習Python 3的string和list |
| 9 | 科學程式設計 | 物理與程式設計，等速一維運動 |
| 10 | 科學程式設計 | 物理與程式設計，拋體運動 |
| 11 | 期中測驗 | 進行程式前測、科學前測、科學運算學習態度前測 |
| 12 | 科學程式設計 | 數學與程式設計，碎形圖與模式辨識 |
| 13 | 科學程式設計 | 數學與程式設計，函式與遞迴實作謝爾賓斯基三角形 |
| 14 | 科學程式設計 | 數學與程式設計，學生自創碎形圖並用程式實作出來 |
| 15 | 科學程式設計 | 生物與程式設計，認識DNA與list語法 |
| 16 | 科學程式設計 | 生物與程式設計，以尼德曼翁施演算法進行DNA序列比對 |
| 17 | 專題製作 | 科學程式設計專題製作與討論，學生在課程中決定期末要製作的主題 |
| 18 | 專題發表+期末測驗 | 1. 學生上台發表專題並演示  2. 進行程式後測、科學後測、科學運算學習態度後測等測驗 |

本次實驗學習者為理學院背景學生，且大多為程式設計初學者。學習者在程式學習過程中，以實作科學運算練習解決程式問題，並透過程式設計建模平台的輔助，在開始寫程式之前先練習程式設計建模的現象描述、資料建模、流程建模、程式化、觀察與除錯等五個步驟。他們在課程中學習Python程式設計的基本語法概念，包含變數(Variable)、判斷式(if, else, else if)、迴圈(for, while)、函式(Function)、清單(List)等內容後，於期中考周進行程式設計測驗作為學生們的程式設計學習前測成績。接著在期中考後進行科學程式設計課程，在科學程式設計課程中，學生以經歷程式設計建模程序的方式來練習解科學運算的程式問題，教師依照課程教材引導學生熟悉程式設計建模的每個步驟，學生必須將自己有關程式解題的想法在程式設計建模平台上寫下，並完成該課程主題的科學程式。學生在最後一週進行專題報告以及程式設計期末考作為學生們的程式設計學習後測成績。

1. 研究工具

本研究選用Python作為主要教學的程式語言，許多研究證實了Python的易學與方便性，作為新手入門文字式程式來說相當合適，導入VPython模組後方便進行視覺化的程式設計模擬。因此本研究以Python作為程式教學目標，所使用之研究工具包含：科學程式設計教材、程式設計學習輔助平台、程式設計能力測驗、科學能力測驗以及態度問卷。以下分別詳述。

**一、科學程式設計教材**

本研究依據建模程序設計建模導向之教材，如圖 3所示。教材內容以科學作為課程主題來引導學生學習相對應的程式概念，學生須從課程中的引導思考將問題表徵，以利接下來將問題程式化。

|  |  |
| --- | --- |
| (a) 物理一維等速運動 | (b) 生物DNA序列 |

圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑4 建模導向之科學程式設計教材(a) (b)

**二、科學程式設計建模平台**

本研究使用之科學程式設計建模平台為研究者自行開發之學習平台，平台上有對應建模程序的功能，如圖 4所示，包含：現象描述、資料建模、流程建模、程式化以及觀察與除錯。程式設計建模輔助平台配合課程教材使用，讓學生在學習科學程式設計時，能透過此程式設計建模程序釐清自己在程式設計時的邏輯與概念。以下敘述平台上主要功能。

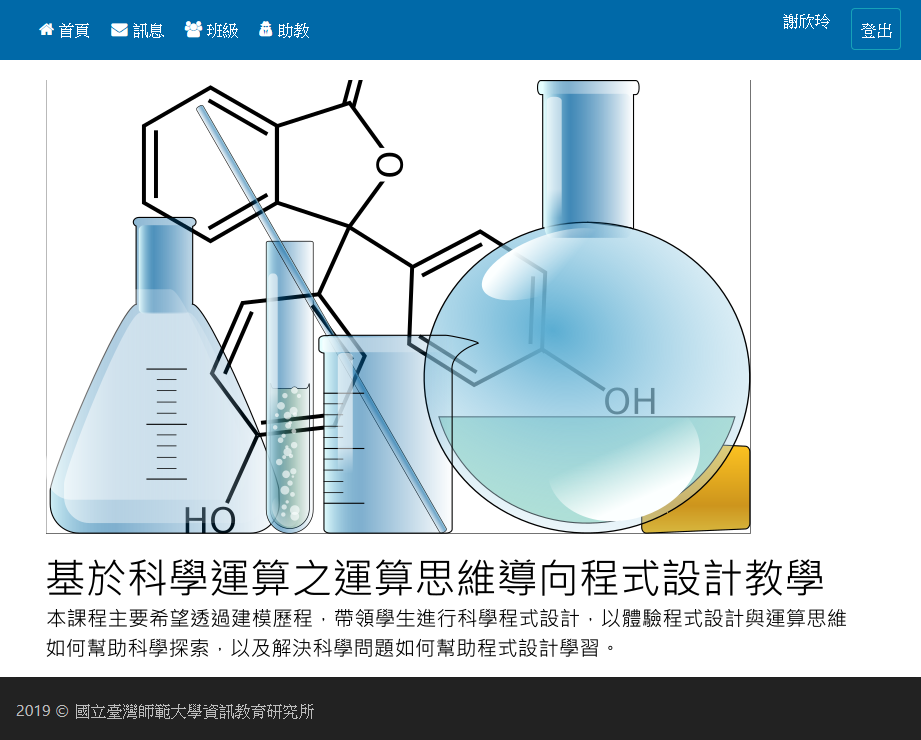


圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑5 程式設計建模輔助平台

**1. 現象描述**

平台的現象描述功能如圖 5所示，目的是引導學生針對待解決的程式問題去思考寫出該程式所需考慮的要素，學生經由將主要的大問題分解成小項目後，再依據這些小項目將自己的想法記錄下來以釐清寫出整個程式之前需要關注哪些面向。



圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑6 程式設計建模輔助平台之現象描述

**2. 資料建模**

平台的資料建模的功能如圖 6所示，目的是讓學生能夠將原本在現象描述中提出的想法轉為程式所需的變數，因此學生需要詳細描述這些變數之間的關係，平台上提供了變數、符號等方塊，讓學生透過建立這些變數通則幫助其後續寫程式。



圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑7 程式設計建模輔助平台之資料建模

**3. 流程建模**

平台的流程建模功能如圖 7所示，目的是讓學生能將原本在現象描述中提出的內容轉為撰寫程式的流程，學生能夠透過流程建模中的流程方塊，整理自己寫出程式前每個步驟的脈絡與邏輯。



圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑8 程式設計建模輔助平台之流程建模

**4. 程式化**

平台的程式化功能如圖 8所示，學生在經歷現象描述、資料建模、流程建模的歷程過後，將對主程式的想法以程式設計來實踐。



圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑9 程式設計建模輔助平台之程式化

**5. 觀察與除錯**

平台的觀察與除錯功能如圖 9所示，學生寫出並執行程式後，若程式執行錯誤，可以回頭觀察程式建模程序中的問題，反覆驗證並修改。



圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑10 程式設計建模輔助平台之觀察與除錯

**三、程式設計能力測驗**

為了檢驗學生對Python的基礎語法的理解程度，研究者選擇課程內提及的相關語法概念為出題核心，重新編成程式設計能力測驗的前測試題，共12題，題目包含：1-8題基礎題，9-12題進階題，並在第11週進行施測，題目如附錄一所示。接著在科學與程式設計一學期的課程結束後，研究者在第18週時進行程式設計能力後測，題型為實作題共有4題，出題內容以測驗學生的程式設計邏輯與程式語法應用為主，題目如附錄二所示。

**四、科學能力測驗**

為了檢驗學生對基礎科學知識的理解程度，研究者選擇課程中提及的科學概念編成科學能力測驗前測試題，題型為應用題共有4題，出題內容包含：物理一維等速運動、理想氣體方程式、生物DNA序列、數學遞迴，題目如附錄二所示，在第11週時進行科學能力前測。接著在科學與程式設計一學期的課程結束後，研究者在第18週時進行科學能力後測，題型為應用題共有4題，出題內容包含：人口成長模型、物理拋體運動、生物DNA序列、數學遞迴，題目如附錄二所示。

**五、科學運算學習態度問卷**

為了理解學生們在科學程式設計課程中的學習態度，研究者在實驗前測與實驗後測分別施行一次科學運算學習態度問卷。本次問卷內容以李克特氏五點量表的形式出題，問卷中包含反向題，每個選項有對應分數(非常同意5分、同意：4分、普通：3分、不同意：2分、非常不同意：1分)，共有28題，題型分為建模經驗、自我效能、學習態度及學習興趣等四大面向，每個大面向又可再細分為其他小項。前測問卷題目如附錄三所示，1-10題為建模經驗(1-2現象描述、3-4流程建模、5-6資料建模、7-8程式化、9-10觀察與除錯)，11-18題為自我效能(11-14程式設計自我效能、15-18科學自我效能)，19-24題為學習態度(19-20程式設計學習態度、21-22科學學習態度、23-24科學程式設計學習態度)，25-28題為學習興趣(25-26程式設計學習興趣、27-28科學學習興趣)。而後測之問卷題目內容與前測大致相同，另外增加2題開放式問答題，以利瞭解學生對課程以及建模步驟的想法，如附錄三所示。

**六、建模學習程序評分標準**

本研究依照文獻整理以及先前前導實驗訂定出此評分標準，對學生在平台上的各項建模能力進行評分。評分標準如表 2所示。

表 2 建模學習程序評分等級與評分標準表

|  |  |  |
| --- | --- | --- |
| 程式建模程序 | 完成項目(分數) | 評分標準描述 |
| 現象描述 | 5 | 1. 學生能知道問題的重點，以繪圖或文字的方式大致描述欲模擬的現象 2. 學生能知道問題的重點，以繪圖或文字的方式詳細描述欲模擬的現象 3. 學生能抽取或分析出問題所包含的基本元素 4. 學生能理解與解釋每個元素的涵義 5. 學生能描述各元素之間的相互關聯 |
| 4 |
| 3 |
| 2 |
| 1 |
| 0 |
| 建立模型  資料建模 | 5 | 1. 學生能辨識或推論出解題的程式結構 2. 學生能描述此科學程式結構的運作 3. 學生能解釋此科學程式結構的涵義 4. 學生能找出解題所需的元素 5. 學生能建立解題所需的規則 |
| 4 |
| 3 |
| 2 |
| 1 |
| 0 |
| 建立模型  流程建模 | 5 | 1. 學生能以流程建模大致說明解題的流程 2. 學生能以流程建模詳細說明解題的流程 3. 學生能以虛擬碼說明解題的流程 4. 學生能辨識或推論出解題的演算法 5. 學生能建立解題所需的規則 |
| 4 |
| 3 |
| 2 |
| 1 |
| 0 |
| 程式與解釋 | 5 | 1. 學生能將模型中各元素以程式變數進行宣告 2. 學生能將選用的程式結構轉換為程式碼，並完成一個以上之程式結構 ( if , for , while ) 3. 學生能將選用的程式結構轉換為程式碼，並完成兩個以上之程式結構 ( if , for , while ) 4. 學生所撰寫的程式能正確運作以顯示欲模擬的現象 5. 學生所撰寫的程式碼能考量一般化或程式風格 |
| 4 |
| 3 |
| 2 |
| 1 |
| 0 |

1. 資料蒐集與分析

**一、資料蒐集**

本研究以科學運算作為研究主軸，欲探討運算思維導向之程式設計教學對學生在程式設計、科學及科學運算學習態度上的學習成效與影響。因此研究者蒐集學生在實驗課程開始前與實驗課程結束後的程式設計、科學及科學運算學習態度的前測與後測成績，以及學生在學期末製作的專題表現作為量化資料；研究者也蒐集學生每次課程中在科學程式建模平台上練習的建模程序，以及他們在科學運算學習態度中的問答題內容作為質性資料。

1. **量化資料**

研究者在實驗開始前進行程式設計、科學及科學運算學習態度的前測。程式設計題目出題內容涵蓋Python程式設計的各項基本語法與邏輯，以檢驗學生在實驗開始前的程式設計能力；科學題目出題內容以實驗內容提及的科學單元作為主，以檢驗學生在實驗開始前的科學能力；科學運算學習態度則是以五點量表的計分方式，將問題分為建模經驗、自我效能、學習態度及學習興趣等四個面向，以檢驗學生在實驗開始前的科學運算學習態度，這些測驗的分數將分別作為學生的程式設計、科學及科學運算學習態度的前測成績。

研究者在實驗最後一次課程時讓學生進行期末專題展示，學生須說明自己專題報告的設計理念、建模程序並且展示程式執行的結果，研究者蒐集學生專題的題目難易度、專題說明書、專題程式碼、專題程式建模及專題總成績等五個面向，作為其專題表現的量化資料。接著進行程式設計、科學及科學運算學習態度後測的測驗，用以檢驗學生在經歷數週科學程式設計課程之後的程式設計能力、科學能力以及科學運算學習態度，出題內容與前測題目所考量的內容大致相同，其測驗的分數則分別作為學生們的程式設計、科學及科學運算學習態度後測成績。

1. **質性資料**

本研究蒐集的質性資料為學生平台建模程序以及態度問卷之問答題。研究者提出了科學程式設計建模程序，這是一項適用於科學領域學生的程式設計的學習方式，我們的每個實驗課程中都會有對應的科學程式設計問題，本研究的目的是讓學生能透過程式設計建模程序逐步思考問題並找出問題的解題方式，而非看到問題就直接開始寫程式。因此在科學程式建模平台上，研究者將每次課程中的程式問題根據程式設計建模程序的步驟依序給予學生解題的指引，並且蒐集學生在平台上填寫的內容為後續資料分析使用。而科學運算學習態度後測問卷中，除了五點量表的題目之外，另外還有兩題開放式問答題，研究者蒐集學生的作答內容，以理解學生對於實驗課程中的程式設計學習方式的想法與感受，以及科學程式建模平台中建模程序如何影響他們對程式設計的思考。

**二、資料分析方法**

研究者根據學生於科學程式設計建模平台上的建模程序依據表6的建模學習程序評分標準評分後，得出班級建模之平均分數為24.59分作為分數界線，接著將學生依照建模品質的高低，在平均分數之上的學生分為建模高品質組(共16人)，他們擁有較好的建模能力；在平均分數之下的學生分為建模低品質組(共16人)，他們較不擅長建模。依照此分組結果來分析兩組學生在程式設計測驗、科學測驗以及學習態度的表現，探討建模品質高低對學生在科學與程式設計課程中的程式設計、科學學習成效差異與學習態度的影響。

1. **建模品質對程式設計學習成就之影響**

研究者將建模能力較好與建模能力較差的學生在程式設計前測與程式設計後測的成績做ANCOVA統計分析，比較兩組學生在經歷科學程式設計課程後學習程式設計的表現是否有顯著差異。

1. **建模能力高低對科學學習成就之影響**

研究者將建模能力較好與建模能力較差的學生在科學前測與科學後測成績做ANCOVA統計分析，比較兩組學生在經歷科學程式設計課程後學習科學的表現是否有顯著差異。

1. **建模能力高低對專題成就之影響**

研究者將建模能力較好與建模能力較差的學生專題表現分為題目難易度、專題說明書、專題程式碼、專題程式建模及專題總成績等五個面向，透過獨立樣本t檢定統計分析，比較兩組學生在經歷科學程式設計課程後專題表現是否有顯著差異。

1. **建模能力高低對科學運算學習態度之影響**

研究者將建模能力較好與建模能力較差的學生在科學運算學習態度前測與後測的成績做ANCOVA統計分析，分為比較兩組學生在經歷科學程式設計課程後的科學運算學習態度是否有顯著差異。科學運算學習態度的前測與後測題目共分為：建模經驗現象描述、建模經驗流程建模、建模經驗資料建模、建模經驗程式化、建模經驗觀察與除錯、程式自我效能、科學自我效能、程式學習態度、科學學習態度、科學程式學習態度 、程式學習興趣、科學學習興趣等共十二個面向。

分析結果與討論

第四章分析結果與討論共分為兩節，第一節說明本研究之實驗資料分析結果，並依據分析結果解釋數據的涵義；第二節探討有哪些因素造成其結果的差異，以及學習者在實驗過程中的變化。

1. 分析結果

**一、 基於建模的程式設計教學對程式設計學習成就之影響**

1. **建模品質對程式設計學習成就之影響**

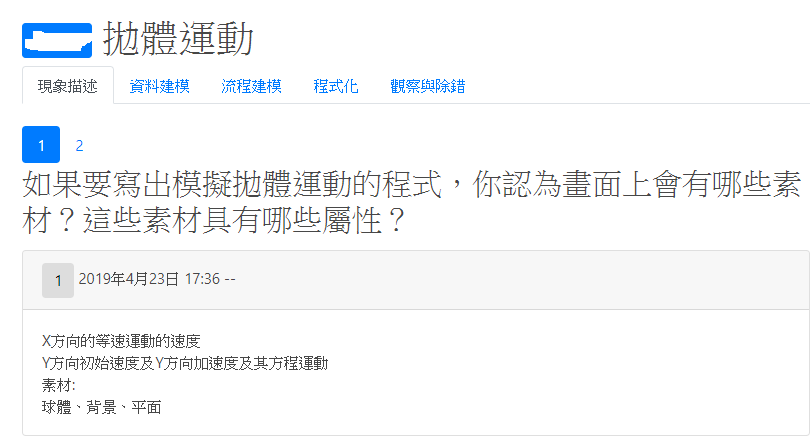
將學生的程式設計期末成績分析，以程式設計期中成績為共變數，程式設計期末成績為依變數，作迴歸斜率同質性檢定，其中 *p* = .107 (> .05)，表示不拒絕虛無假設，兩組資料的誤差變異量並沒有顯著差異，具有同質性，因此可進行ANCOVA分析。

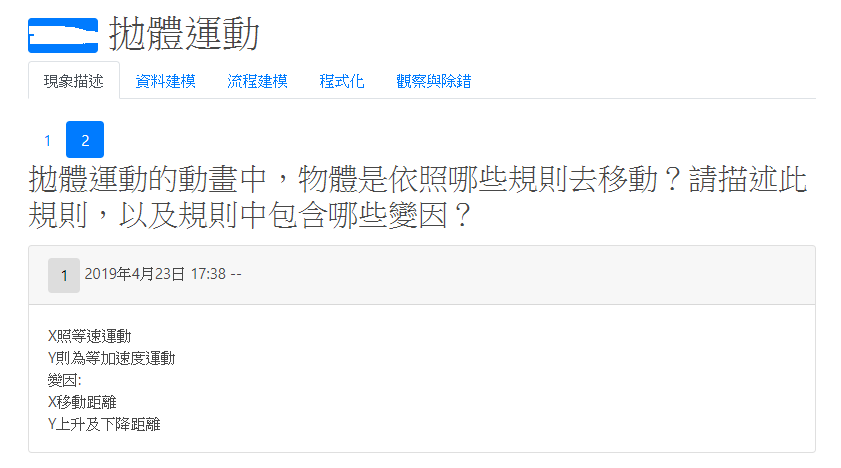
在描述性統計中，建模能力較好的16人於程式學習成就平均為15.31，標準差為3.30；建模能力較差的16人於程式學習成就平均為10.88，標準差為4.69。在ANCOVA分析中，兩組學生之顯著性為.016 (*p* < .05)，表示建模品質較好的學生其程式設計學習成就顯著高於建模品質較差的學生。

表 3 建模高低品質組之程式設計學習描述性統計資料及ANCOVA分析

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 建模品質 | 人數 | 前測 | | 後測 | | *F* | *p* |
| *M* | *SD* | *M* | *SD* |
| 高 | 16 | 14.06 | 5.56 | 15.31 | 3.30 | 6.484 | .016\* |
| 低 | 16 | 10.56 | 7.20 | 10.88 | 4.69 |

研究者對其程式設計學習成就進行ANCOVA分析後，發現建模品質較好的學生在程式設計學習的成效較建模品質較差的學生好，其中科學程式建模平台在輔助學生程式設計上具有影響力，圖 10為建模品質較好的學生於本研究課程中物理與程式設計單元的程式設計建模模型。由模型特徵來看，建模品質較好的學生在課程中完成的程式模型內容整體較為完整，能夠依照程式設計建模程序寫出符合題目需求的內容，而建模品質較差的學生在程式設計建模平台上寫下的內容大多不完整或留白。





(a)現象描述(1.2題)



(b)資料建模



(c)流程建模



(d)程式化

圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑11 建模高品質組學生的程式設計建模(a)(b)(c)(d)

**二、 基於建模的程式設計教學對科學學習成就之影響**

**1. 建模品質對科學學習成就之影響**

將學生的科學期末成績分析，以科學期中成績為共變數，科學期末成績為依變數，作迴歸斜率同質性檢定，其中 *p* = .759 (> .05)，表示不拒絕虛無假設，兩組資料的誤差變異量並沒有顯著差異，具有同質性，因此可進行ANCOVA分析。

在描述性統計中，建模能力較好的16人於科學學習成就平均為17.31，標準差為1.66；建模能力較差的16人於程式學習成就平均為15.87，標準差為3.76。在ANCOVA分析中，兩組學生之間的顯著性*p* = .273 (> .05)，表示建模品質較好的學生其科學學習成就與建模品質較差的學生之間未達顯著差異。

表 4 建模高低品質組之科學學習描述性統計資料及ANCOVA分析

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 建模品質 | 人數 | 前測 | | 後測 | | *F* | *p* |
| *M* | *SD* | *M* | *SD* |
| 高 | 16 | 15.25 | 2.02 | 17.31 | 1.66 | 1.248 | .273 |
| 低 | 16 | 13.88 | 2.80 | 15.87 | 3.76 |

建模品質較好與建模品質較差的學習者於科學學習成就上的表現無差異，如上述表所示。由於學習者們都熟悉題目背後的科學知識，課程也以科學領域重視的建模來作為學習程式的工具，結果與原本預期學習者在此項目上的表現會有差異的研究假設結果不同。

我們在研究中以科學主題作為程式設計課程的主軸，讓學生透過實作科學運算來進行現象的模擬，許多學生提到了自己一開始對學習程式設計的緊張，也有不少學生提到以自己熟悉的科學主題來進行程式設計學習能讓自己更容易掌握，並且認為課程中透過科學運算學習程式設計的方式抱持肯定的想法的學生佔了總人數的71.88%，如表 5所示。

表 5 學生對以科學運算來學習程式設計的想法

|  |  |
| --- | --- |
| 請問在這學期的課程中，透過解決科學問題的方式來學習程式設計，是否讓你更容易學程式設計？為什麼？ | |
| No. | 學生回答 |
| 5 | 會比較有趣，主題比較明確 |
| 6 | yes, because it's help me easier to understand |
| 8 | 感覺比較不同，單純的程式設計可能著重要程式的技巧上，而這種感覺是從問題出發去學習 |
| 9 | 是,先從熟悉的事物來學習不熟的事較能上手 |
| 11 | 是，因為是一步一步將問題拆解，並將解決方法的脈絡釐清後，循著脈絡去寫出對應的程式碼，所以我覺得透過解決科學問題的方式來學習程式設計讓我更好上手 |
| 15 | 因為了解那些現象的運作原理後，在轉換成程式語言會比較簡單 |
| 16 | 還好，但是好險有學Snakify的教學讓我有python的基礎。因此後來看科學問題時比較能從程式設計的角度觀察。 |
| 17 | 是,透過實際的科學例子,讓程式碼不再是一堆未知文字,而能知曉其中意涵 |
| 18 | 多少有點幫助(應該說更能知道現實生活的問題是可以用程式解決的) |
| 19 | 是，因為有固定的步驟去引導 |
| 21 | 因為我是理組的 |
| 23 | 是，能夠讓我更容易分析科學問題，進而各個突破！ |
| 24 | 是。將科學問題寫成程式時，遇到的問題絕對不會比單純計算、求數學解答的少，這對於練習很有幫助。 |
| 26 | 是，因為透過一步步的步驟，讓我更了解其中的過程 |
| 27 | 是，能釐清問題重點 |
| 29 | 有的，一般來說寫程式最擔心就是太抽象的概念。若從科學的角度去著手，會比較有一個模型或想法。 |
| 31 | 是，讓我有更多的想法 |
| 32 | 是，更加具體的認知程式碼的作用 |
| 33 | 是的 原本零基礎的我 以前都對程式設計這東西非常陌生也感到很困難 因為這堂課更知道程式設計如何運用 |
| 34 | 我覺得科學問題跟程式設計是同性質的東西，都需要邏輯，都需要數學，所以藉由解決科學問題來學習程式，更好上手 |
| 35 | 是、因為我本身就是學習科學的人，很能夠理解科學問題以及各類科學模型，但是程式是一個完全未知的領域，不太知道要怎麼把現實生活中的變量轉換成程式的形式，透過這個課程，能讓我更能理解科學與程式之間的關聯，怎麼把簡單的科學模型程式化 |
| 37 | 是，以實際的例子進行架設與說明，使程式有視覺化的成果，更為清晰的能理解其中影響的部分 |
| 38 | 是，因為在用此方式時，可以將一些想不到的地方，化解開來。 |

**三、 基於建模的程式設計教學對專題成就之影響**

**1. 建模品質對專題表現之影響**

建模能力較好的學生在題目難易度、專題說明書、專題程式碼、專題程式建模、專題總成績五個面向的顯著性分別為 .005、 .000、 .010、 .045、 .000皆< .05，表示建模品質較好的學生其專題表現整體成效顯著高於建模品質較差的學生。

表 6 建模高低品質組之專題表現描述性統計資料及T-test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 面向 | 組別 | *M* | *SD* | *t* | *p* |
| 題目難易度 | 建模高品質組a | 8.063 | 1.289 | 3.021 | .005\* |
| 建模低品質組b | 6.125 | 2.217 |
| 專題說明書 | 建模高品質組a | 25.000 | 3.483 | 5.665 | .000\*\*\* |
| 建模低品質組b | 16.875 | 4.558 |
| 專題程式碼 | 建模高品質組a | 26.063 | 3.295 | 2.776 | .010\* |
| 建模低品質組b | 23.188 | 2.509 |
| 專題程式建模 | 建模高品質組a | 15.938 | 4.611 | 2.101 | .045\* |
| 建模低品質組b | 13.000 | 3.162 |
| 專題總成績 | 建模高品質組a | 75.063 | 7.388 | 5.063 | .000\*\*\* |
| 建模低品質組b | 59.068 | 10.133 |
| a*n* = 16. b*n* = 16. *p* < .05. *p*\*\*\* < .000 | | | | | |

研究者比較了學生在專題上的各項能力，建模品質較好的學生的專題製作書內容示例如圖 11所示，建模品質較差的學生的專題製作書內容示例如圖 12所示。在題目難易度的部分，題目難度分數滿分為10分，研究者將6分以上定為困難題型的標準，學生們選擇的專題難度如圖 13所示，建模品質較好的學生選擇的專題主題大多以探索科學或數學問題為主，如：司乃爾定律、行星運行，建模品質較好的學生中選擇困難題型的人數佔93.75%；建模品質較差的學生選擇的題目大多難易度偏簡單，如：找三角形、找質數，甚至有學生直接使用科學程式設計課程中的範例教材來完成專題。在專題說明書的部分，建模品質較好的學生會使用較多的資料以及描述他們如何發想專題的面向去介紹他們的專題；建模品質較差的學生對自己專題的描述與介紹則較為簡略。在專題程式碼的部分，建模品質較好的學生在其專題中使用較多科學程式設計課程教過的語法，其程式的結構與複雜度也較高，此外研究者也觀察到有許多學生會在程式碼中作註解；建模品質較差的學生使用的程式語法與撰寫的程式結構都較簡易，會在程式碼中作註解的比例也較低。在專題程式建模的部分，建模品質較好的學生在他們的專題程式設計建模內容完成度較高；建模品質較差的學生在專題程式設計建模的內容完成度相對較低。

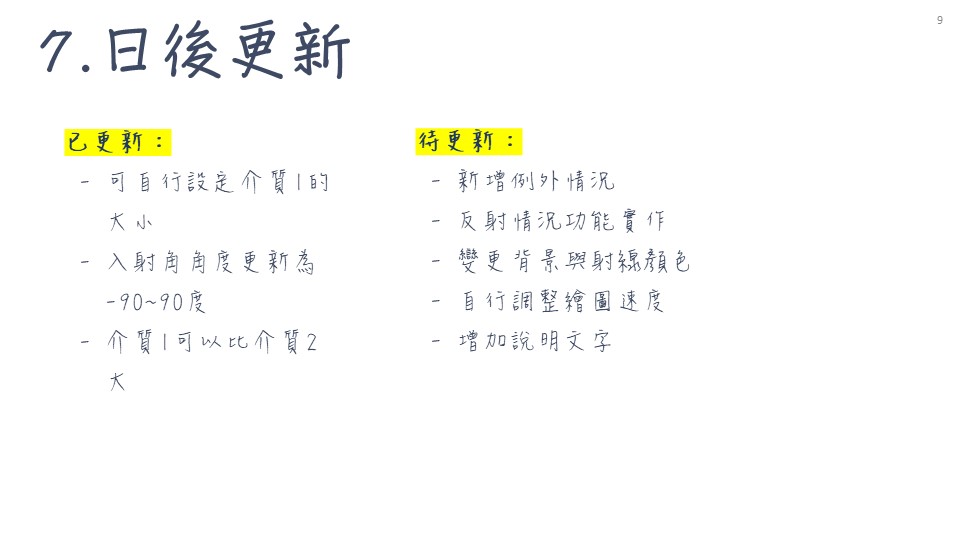
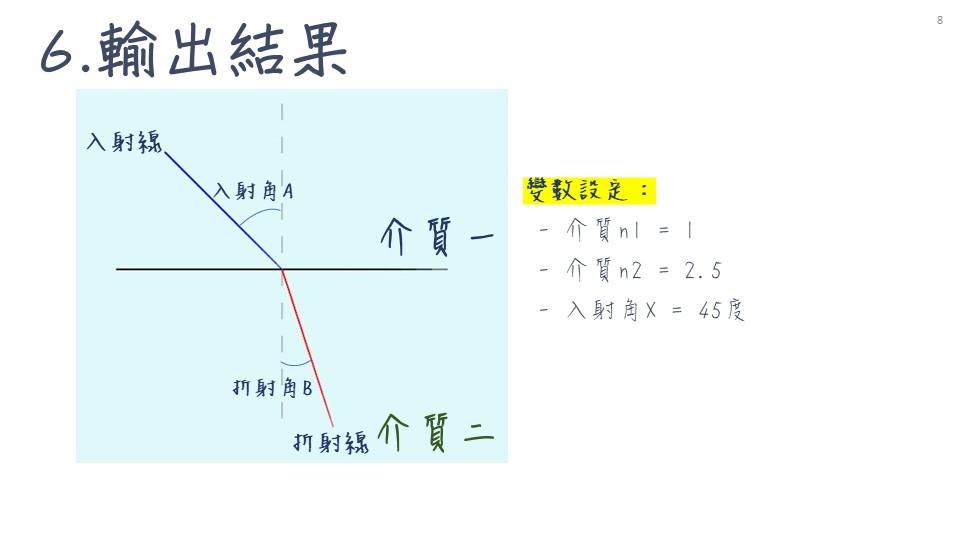
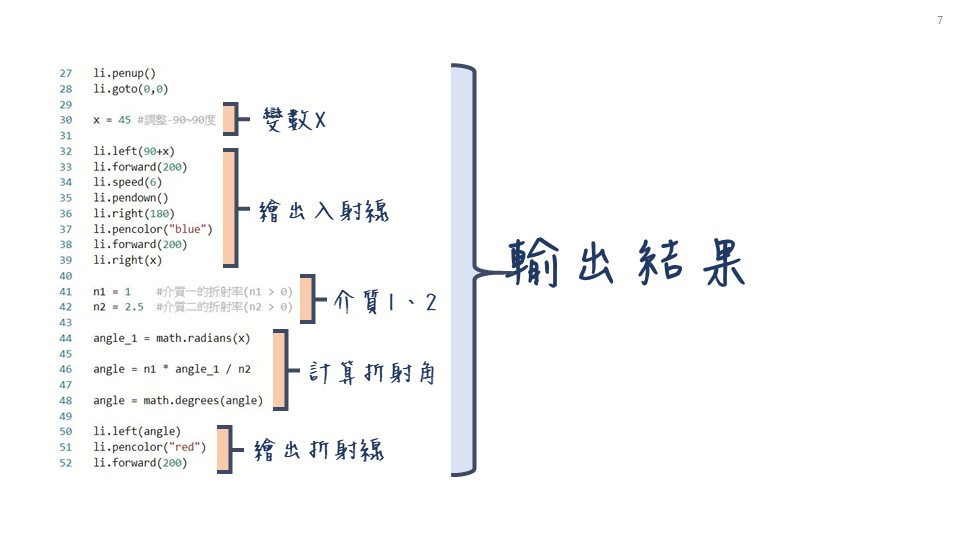
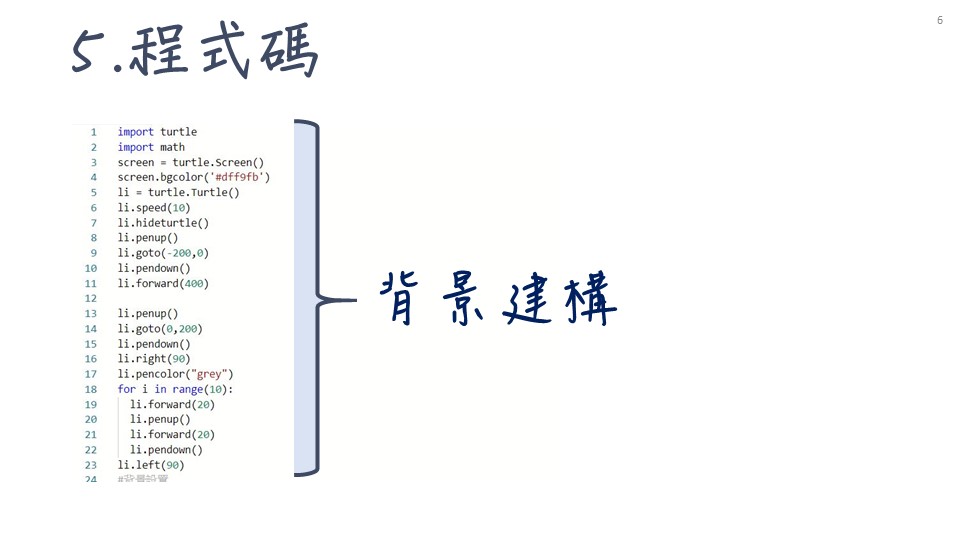
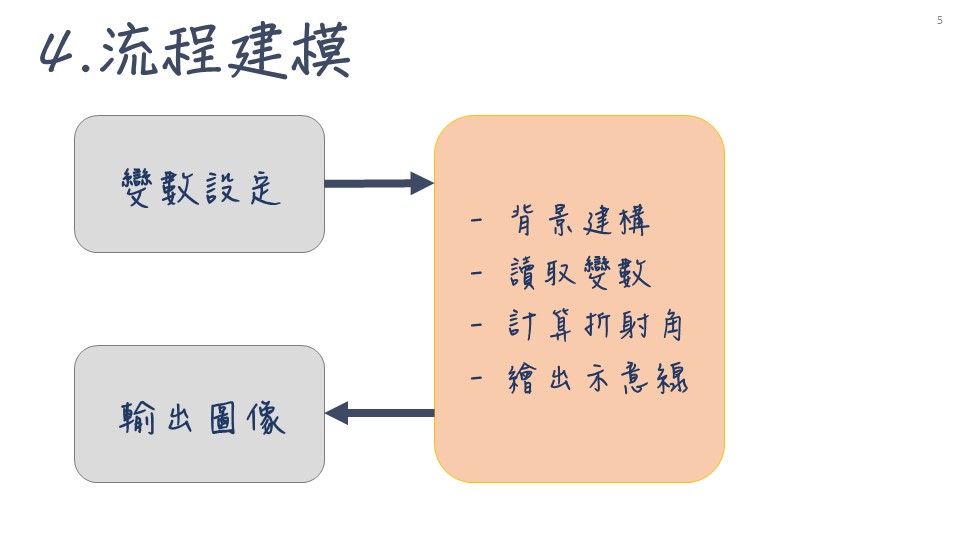
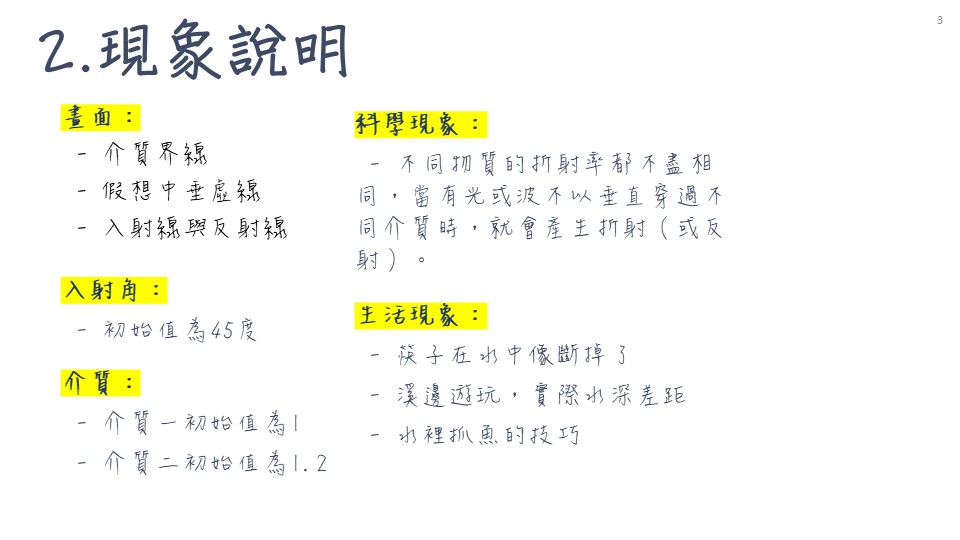
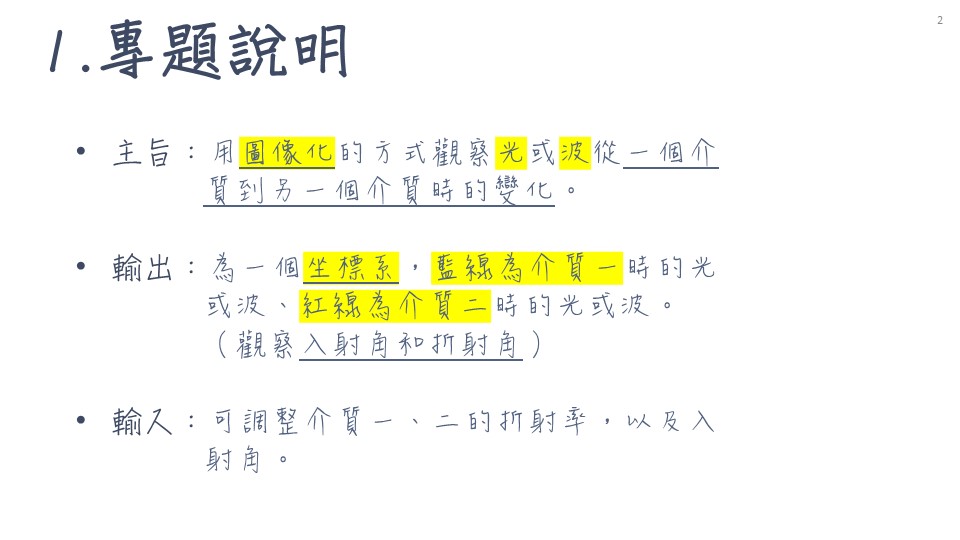


圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑12 建模高品質組學生的專題說明書示例(主題：司乃爾定律)

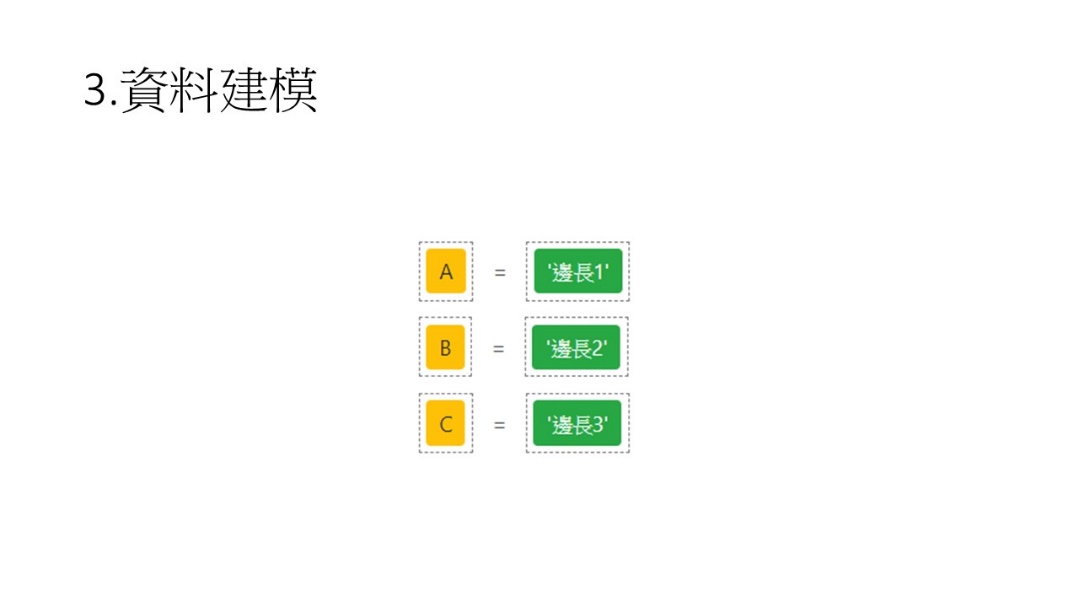
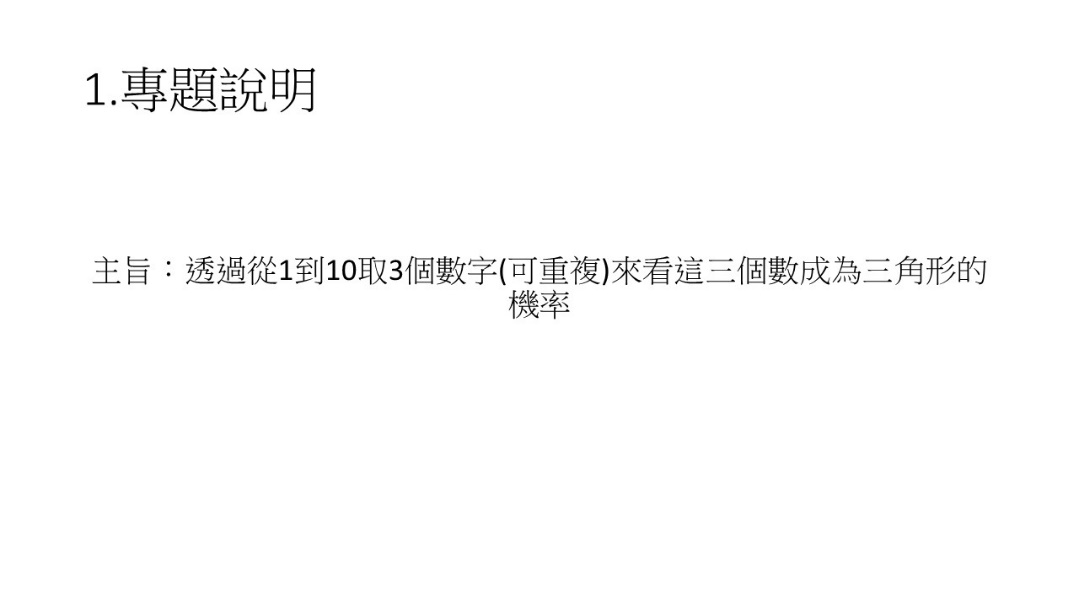


圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑13 建模低品質組學生的專題說明書示例(主題：找三角形)

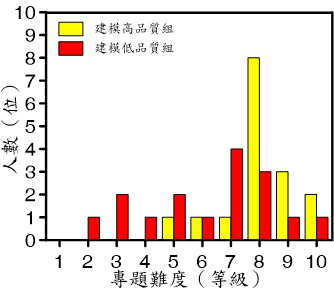


圖 **錯誤! 所指定的樣式的文字不存在文件中。**‑14 建模高品質組學生與建模低品質組學生選擇專題難度的人數

**四、 基於建模的程式設計教學對科學程式學習態度之影響**

**1. 建模品質對科學程式學習態度之影響**

將學生的期末學習態度成績分析，以學習態度期中成績為共變數，學習態度期末成績為依變數，作迴歸斜率同質性檢定。

其中在迴歸斜率同質性檢定之建模經驗現象描述、建模經驗流程建模、建模經驗資料建模、建模經驗程式化、建模經驗觀察與除錯、程式自我效能、科學自我效能、程式學習態度、科學學習態度、科學程式設計學習態度 、程式學習興趣、科學學習興趣*p*值分別為： .098、 .008、 .064、 .502、 .809、 .963、 .404、 .378、 .191、  
 .010、 .576、 .189，除了建模經驗流程建模與科學程式設計學習態度之外的其餘十項*p* > .05，表示這十項資料皆不拒絕虛無假設，資料的誤差變異量並沒有顯著差異，具有同質性。由於本研究分析採用等組設計，建模能力較好與建模能力較差這兩組的學生皆為16人，對於違反常態性和變異數同質性等基本假設，變異數分析仍具有其強韌性，不太會影響第一與第二類型錯誤機率，因此這十二個面向均可繼續進行ANCOVA分析。

由ANCOVA分析可看出建模能力較好的學生與建模能力較差的學生，在建模經驗觀察與除錯中有差異，在其他項目上則沒有差異，如表 7所示。學生建模能力的好壞會影響他們如何對整個程式設計建模程序中發生的問題進行思考，並能反覆修正程式與模型中的問題。由科學程式設計建模平台上觀察與除錯的步驟來看，這項步驟在學生們完成程式設計建模程序後進行，當學生們依照自己在程式設計建模程序完成的程式未順利執行時，他們可以與師長及同儕討論自己程式的問題、觀察自己程式碼中的邏輯或語法錯誤，或者回頭檢驗自己當初撰寫的程式設計建模模型。

表 7 建模品質高低之期末科學程式學習態度描述性統計資料及ANCOVA分析

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 描述性統計 | | | | | | | | ANCOVA | |
|  | 前測 | | | | 後測 | | | | *F* | *p* |
|  | 建模高品質組a | | 建模低品質組b | | 建模高品質組a | | 建模低品質組b | |
|  | *M* | *SD* | *M* | *SD* | *M* | *SD* | *M* | *SD* |
| 建模經驗現象描述 | 8.19 | 1.17 | 7.69 | 1.08 | 8.44 | 1.21 | 8.50 | .89 | .274 | .605 |
| 建模經驗流程建模 | 8.25 | 1.13 | 7.94 | 1.24 | 8.13 | 1.36 | 8.31 | .95 | .647 | .428 |
| 建模經驗資料建模 | 8.31 | 1.14 | 7.81 | 1.33 | 7.31 | 1.62 | 7.31 | .95 | .001 | .971 |
| 建模經驗程式化 | 6.69 | 1.35 | 6.94 | 1.48 | 8.13 | 1.15 | 8.31 | 1.01 | .121 | .730 |
| 建模經驗觀察與除錯 | 7.69 | 1.30 | 7.75 | 1.48 | 7.19 | 1.17 | 6.25 | 1.06 | 5.655 | .024\* |
| 程式設計自我效能 | 15.38 | 2.50 | 14.63 | 2.22 | 15.19 | 2.99 | 14.38 | 2.66 | .083 | .775 |
| 科學自我效能 | 16.19 | 2.23 | 14.75 | 2.02 | 15.69 | 2.55 | 15.44 | 2.00 | 3.745 | .063 |
| 程式學習態度 | 9.25 | 1.00 | 8.69 | 1.35 | 8.94 | 1.48 | 8.63 | 1.09 | .028 | .868 |
| 科學學習態度 | 8.63 | 1.41 | 8.13 | .96 | 9.00 | 1.15 | 8.63 | .89 | .103 | .751 |
| 科學程式設計學習態度 | 8.75 | 1.34 | 8.31 | 1.25 | 8.94 | 1.34 | 8.56 | .89 | .229 | .636 |
| 程式學習興趣 | 8.81 | 1.05 | 8.19 | 1.22 | 9.31 | .95 | 8.75 | 1.13 | .629 | .434 |
| 科學學習興趣 | 8.31 | 1.25 | 8.00 | 1.26 | 8.88 | 1.26 | 8.38 | 1.31 | .795 | .380 |
| a*n* = 16.  b*n* = 16. *p*\* < .05 | | | | | | | | | | |

研究者在科學程式設計學習態度問卷中調查學生對學習平台建模步驟分項及整體的看法，如表 8所示。學生認為在進行程式設計建模程序時，透過現象描述能有助於自己對問題的理解，可以釐清在開始寫程式之前需要做哪些準備，並將自己最初步的想法先記錄下來。透過流程建模能夠制訂程式設計撰寫的流程，列出每一個步驟要執行的內容。透過資料建模能夠從中找出符合題目的參數與概念，知道接下來在程式中需要使用哪些資料。整體而言，他們認為運用程式設計建模平台的建模步驟比起直接動手寫程式，經由逐步的引導更有助於他們對程式整體的理解。

表 8 學生對使用平台建模程序習程式設計的看法

|  |  |  |
| --- | --- | --- |
| **請問科學程式設計平台中的哪些步驟，對你學習程式設計是有用的？為什麼？** | | |
| 步驟 | 編號 | 學生回答 |
| 現象描述 | 8 | 我覺得是第一步是最重要的，需要釐清是什麼 |
| 23 | 現象描述，在解析的過程中往往能讓我找到解決的辦法! |
| 29 | 現象描述讓我把預想中要做的事情都記錄下來。 |
| 38 | 現象描述，因為有時候會忘記一些重要的事情，但當了解了架構後，在寫後面的程式時，比較不容易出錯，也比較能夠清楚整個程式 |
| 流程建模 | 9 | 流程建模,能把腦袋想的依序列出來才知道那裡需要修改 |
| 14 | 流程建模，清楚解決問題的過程 |
| 16 | 我覺得流程建模有用，因為能清楚的知道你每個步驟的目的、想法 |
| 17 | 流程建模與程式碼的部分 |
| 20 | 流程建模，因為可以先釐清順序。資料建模的使用很不便，自由度不高 |
| 資料建模 | 11 | 資料建模，因為在那部分我可以將所有需要的參數或變數一一列出，比較不會有漏掉某個參數的問題發生。 |
| 26 | 資料建模，就像是一個打程式前的一個大綱 |
| 27 | 資料建模，建模後能清楚知道自己要打哪些東西 |
| 31 | 資料建模，能讓我釐清我所需要用到的資料 |
| 33 | 這平台的資料建模我覺得很有用 當中使我們思考 如何建構完整的思考模式 |
| 34 | 資料建模，因為在解決問題前必須先了解有哪些東西需要當參數，還有背後的原理 |
| 36 | 資料建模，模擬寫程式時需要的物件 |
| 其他想法 | 10 | 跟別人解釋程式在做什麼的時候比較容易有脈絡 |
| 13 | 有，把很多麻煩的地方簡化了 |
| 35 | 有用，這個平台比較不用直接打生冷的程式碼，可以先文字化，比較方便理解 |
| 37 | 引導式的一步一步把程式運行脈絡做出，比直接打程式更能理解程式如何運行 |

**2. 科學程式設計學習態度問卷信度分析**

科學學習態度問卷於期中考週及期末專題報告時施測，分為建模經驗現象描述、建模經驗流程建模、建模經驗資料建模、建模經驗程式化、建模經驗觀察與除錯、程式自我效能、科學自我效能、程式學習態度、科學學習態度、科學程式學習態度 、程式學習興趣、科學學習興趣共十二個面向。題目類型有李克特氏五點量表題和開放式問答題。下表為學習態度十二個面向之信度分析結果。

表 9 學習態度十二個面向之信度分析結果

|  |  |  |
| --- | --- | --- |
| 面向 | | α |
| 建模經驗 | 現象描述 | .521 |
| 流程建模 | .569 |
| 資料建模 | .500 |
| 程式化 | .598 |
| 觀察與除錯 | .732 |
| 自我效能 | 程式設計 | .704 |
| 科學 | .691 |
| 學習態度 | 程式設計 | .755 |
| 科學 | .756 |
| 科學程式設計 | .783 |
| 學習興趣 | 程式設計 | .750 |
| 科學 | .693 |
|  | | |

1. 討論

為確保學生在實驗前的程式設計概念一致，在實驗開始前先進行Python程式語言教學，進入正式實驗課程前施行程式設計前測，實驗課程結束後實施程式設計後測以驗證其成效，本研究將學生在實驗課程中的建模程序評分作為他們建模能力高低分組依據。

**一、 建模能力對程式設計學習成就的意義**

科學程式設計課程以專題式學習的學習策略實施，在課程中以程式設計建模來引導學生實踐科學運算進行程式設計與科學的學習。為探討建模能力對學生程式設計學習成就的意義，將學生依照科學程式建模平台中的建模品質分數進行建模高、低品質分組，並將其程式設計前測成績與程式設計後測成績進行ANCOVA分析，分析結果如表 3所示。本研究中該課程的所有學生都接受相同的課程教材與課程教學策略來學習科學程式設計，教學者依據教材內容進行教學，接著學生必須根據課程題目需求來完成程式設計建模以及題目要求之科學程式。研究者發現建模能力較好的學生相較於建模能力較差的學生在程式設計的學習上有較好的表現，以下將就學生的學習情況做討論。

在科學程式模型建立的部份，建模能力較好的學生建立的科學程式模型通常概念較精確、細緻、邏輯清晰，較容易掌握解題的脈絡與解題所需的關鍵要素；建模能力較差的學生建立的科學程式模型通常概念較模糊、零碎、邏輯簡略，較難以看出解題的想法與策略。建模是科學領域用來理解現象的重要技能，透過模型的建立能夠將複雜現象簡化以利思考，使理論容易被理解或進行預測(Gilbert et al., 2000)。在科學程式設計課程中介紹的程式語法包含：變數、迴圈、條件判斷以及序列等概念，對於相同的科學程式設計主題概念，會因為每個人的心智模型不同導致建立的科學程式設計模型也不同。在本研究中，學生們建立的模型品質好壞間接成為影響他們在科學程式設計學習的情形，進而影響他們在程式設計上的表現。程式設計是一門抽象的專業技能，對新手而言學習上容易產生困難，而建模是能夠幫助學生將抽象的概念做出適切表徵的一項學習策略，並且能夠幫助他們釐清自己對問題的理解，將符合題目的概念進一步抽象化後以找出解題所需的知識概念，再將概念之間聯繫起來，完成科學程式設計建模。由此可見，擅長建立高品質模型的學生擁有較好的問題解決策略，在程式設計的學習上更有成效。

**二、 建模能力對科學學習成就的意義**

為探討建模能力對學生科學學習成就的意義，將學生依照科學程式建模平台中的建模品質分數進行建模能力高低品質分組，並將其科學前測成績與科學後測成績進行ANCOVA分析，分析結果如表 4所示。研究者發現建模能力較好的學生與建模能力較差的學生在科學的學習上表現沒有顯著差異，以下將就學生的學習情況做討論。

本研究在科學程式設計課程中所介紹的科學主題包含：物理一維等速運動、物理拋體運動、生物DNA序列以及數學碎形圖。從學生的學習表現觀察，建模能力較好的學生在科學前測成績表現平均為15.25，科學後測成績平均表現為17.88；建模能力較差的學生在科學前測成績平均表現為13.88，科學後測成績平均表現為15.87。本研究主要著重於理學院大學生的程式設計學習，科學單元為輔助其進行跨領域學習之用，原先研究者預期學生能因為建模能力好壞導致其在科學領域的學習成就會產生差異，但科學畢竟是一門需要較長期時間去深入探究的專業科目，實驗課程基於時間有限每單元僅實施2-3週，加上本次實驗的研究對象來自不同科學領域背景，例如：物理系、化學系、生科系、地科系、數學系、電機系等，於是在課程安排上需要兼顧來自不同科系的學生對科學與程式設計課程的需求，從多元的面向設計科際整合的課程讓學生體會科學運算的功用，因此課程設計無法專精在其中一門科學領域上，導致短期內較難以看出科學能力因建模能力好壞而產生的差異。

**三、 建模能力對專題表現的意義**

為探討建模能力對學生專題表現的意義，將學生依照科學程式建模平台中的建模品質分數進行建模高低品質分組，並將其專題學習成績進行獨立樣本t檢定分析，分析結果如表 6所示。研究者發現建模能力較好的學生在專題學習上的各項表現均較建模能力較差的學生好，以下將就學生在專題中的學習情況做討論。

本研究中的專題學習成就分為：題目難易度、專題說明書、專題程式碼、專題程式建模、專題總成績五個面向。研究者觀察兩組學生們在專題中的表現，專題題目為學生自行決定想做的主題，建模能力較好的學生在專題難度上傾向挑戰難度較高較複雜的科學現象作為題目，建模能力較差的學生在選題上的趨勢較為分散，因此專題難易度上有所差異。

在專題說明書的部分，建模能力較好學生的報告內容較豐富，學生們會用較多的篇幅來介紹專題中的科學主題，還有他們在完成專題時的解題思路，此外他們也多半會額外加入自己對專題的反思以及他們還能怎麼改進這個專題程式，讓人感受到這群學生真實經歷了探究科學的過程。建模能力較差的學生報告內容則是較簡略，他們報告書的篇幅大多只完成教學者要求的最低限度，以致於難以從中看出他們在專題中的解題思路與想法。

在專題程式碼與專題程式建模的部分，建模能力較好的學生在科學程式設計建模較完整，他們能夠依據對應他們主題的科學概念進行詳細的現象描述，能精確找出對應他們主題的關鍵要素做資料建模，能夠設計出符合專題所需的專題程式解題程序做流程建模，並且能夠依據上述的歷程將想法轉換為程式碼，所以整體專題程式的完成度較高。相較於此，建模能力較差的學生在科學程式設計建模內容較淺白且不完整，研究者認為學生們是出於他們對自己選擇的專題主題較熟悉，以致於他們在程式設計建模的第一步驟現象描述大多是以大綱式帶過而缺少描述的歷程，資料建模時的資料也並非源自現象描述中的內容或者定義不清，流程建模中也難以看出解題的程序與邏輯，程式碼的程式語法或邏輯往往會有錯誤導致專題程式不完整，這可能是源自學生於程式設計建模階段建立的模型太模糊而造成他們在程式碼轉換上的困難。

**四、 建模能力對科學程式設計學習態度的影響**

為探討建模能力對學生科學程式設計學習態度之影響，將學生依照科學程式建模平台中的建模品質分數進行建模高低品質分組，並將其前測與後測的科學程式設計學習態度問卷進行ANCOVA分析，分析結果如**錯誤! 找不到參照來源。**及**錯誤! 找不到參照來源。**所示。本研究中的程式設計學習態度分為：建模經驗現象描述、建模經驗流程建模、建模經驗資料建模、建模經驗程式化、建模經驗觀察與除錯、程式自我效能、科學自我效能、程式學習態度、科學學習態度、科學程式設計學習態度 、程式學習興趣以及科學學習興趣共十二個面向。研究者發現建模能力較好的學生相較於建模能力較差的學生在建模經驗的觀察與除錯的項目中有較顯著的差異，以下將就學生的學習情況做討論。

從研究者在課堂中的觀察來看，科學與程式設計課程每周進行2節課，教學者會依照科學程式設計教材的主題來帶領學生探索科學程式設計，每單元都有對應的科學主題以及程式設計語法概念，並且有1到3題不等的科學程式設計題，待教學者教學完畢後剩餘的時間會留給學生們完成對應課程的主題程式。學生們在開始寫程式前，習慣花較多時間在程式設計平台上建立對應課程主題的程式設計模型，他們會將現象描述、資料建模、流程建模的步驟都完成後，才開始進行程式化的步驟。由於程式設計建模平台上並未有程式編譯功能，學生必須另外在其他Python的編譯器(例如：IDLE)上寫程式，因此較難掌握學生每次執行程式時遇到的錯誤，不過在課堂觀察中能看到學生們在每次執行程式後，會再回頭修改他們的程式設計模型。儘管建模能力較好與建模能力較差的學生們對於建模程序觀察與除錯的認同降低，但從他們在程式設計學習成效來看，兩組學生的平均成績都有進步如表 3所示，表示學生們雖然沒有感受到程式設計建模程序對他們在程式設計學習上的影響，實質上卻是有助益的。另外，從表 8可以發現學生們對於程式設計建模程序的看法，不論是在現象描述、資料建模、流程建模，這些程序都有學生感到認同。學生們認為現象描述是重要的，開始寫程式前第一步驟要先釐清題目的意義，針對問題先做解題分析，從中找出解題方法後將所有想法先記錄下來，待了解問題解決的架構後，可以對科學程式問題有更全面的理解(8、23、29、38)。流程建模的步驟能將想法條列出來，整理出每一步驟要如何進行，可以更清楚解決問題的歷程，也能容易找出需要修改的部分(9、14、16、17、20)。資料建模的步驟可以將解題所需的參數與變數列出，避免遺漏參數的風險，且能知道每一項目的定義，幫助他們建立對問題更加完整的思考(11、26、27、31、33、34、36)。整體而言，學生們認為透過科學程式建模程序的引導，能夠讓他們在開始進行科學程式設計前先將想法以文字或圖形的方式列出，方便對問題進行抽象化，且有助於對問題的理解，在向他人解釋程式上也更有脈絡能依循(10、13、35、37)。

**五、 綜合討論**

從實驗結果發現，學生建模能力好壞會影響其程式設計表現，也會影響專題表現，而在科學表現上則未看出差異。除此之外，學生建模能力好壞也影響他們對完成程式設計後經歷觀察與除錯的看法。

根據實驗分析結果來看，本研究提出之基於建模的程式設計教學能夠增進學生在程式設計上的學習成效，但對科學的學習成效並沒有影響。在專題表現上，建模能力較好的學生於題目難易度、專題說明書、專題程式碼、專題程式建模、專題總成績五個面向的表現皆比建模能力較差的學生好。

此外，學生經歷基於建模的程式設計教學，對他們在程式設計建模的學習態度中，現象描述、資料建模、流程建模、程式化都不因建模能力好壞而產生差異，在觀察與除錯中則因學生建模品質高低而產生差異。學生在經歷基於建模的程式設計教學後，建模能力較好與較差組別的學生對程式設計的學習興趣、學習態度以及自我效能皆未產生差異，對科學的學習興趣、學習態度以及自我效能也未產生差異。

結論與建議

本研究以科學程式設計的方式帶領學生透過建模學習程式設計，並分別評估其各項成效。為了達成此目的，研究中提出程式設計建模程序並且以教學平台給予學生程式建模練習，也設計適用於理學院學生的科學與程式設計教學課程。本章節將結合前一章節之研究分析結果與討論，給予此種教學模式結論與建議。第五章結論與建議共分為兩節，第一節將整合研究結果統整出結論；第二節則根據未來發展的基於科學運算之運算思維導向程式設計教學給予研究方面的建議。

1. 結論

本節將依據第四章分析結果與討論中的內容，為本次實驗基於科學運算之運算思維導向程式設計教學所提出之研究問題做出整體性的結論。在實驗中，研究者以程式設計建模作為引導理學院大學生學習程式設計的鷹架，設計以此背景為主之學生適用的科學與程式設計課程，並提供平台輔助學生進行程式設計建模程序。研究者根據學生們的測驗成績、專題、問卷量表、平台中建模程序以及課堂學習情況，用以了解學生們在程式設計與科學上的學習成就、學習態度以及專題表現。最後，研究者將蒐集之各項質性與量化資料做分析，並探討此種教學方式對於學生們在運用程式設計建模程序後在各項欲探討項目之影響。依據上述之實驗分析與討論結果，本研究結論如下：

**一、 本研究之學習者透過基於科學運算之運算思維導向程式設計教學後，發現建模品質較好的學生在程式設計學習表現上較佳**

研究者在本次科學與程式設計課程中進行教學實驗，實驗中以程式設計建模程序以及程式設計建模輔助平台來引導理學院大學生探究科學主題並進行科學運算，並且在期中考與期末考週分別進行程式設計前測與後測，將學生依照建模能力高低分組後以ANCOVA分析此數據。依據本次實驗結果發現，學生建模能力好壞會影響其在程式設計學習的成效。由學生在平台上建立的模型來看，建模能力較好的學生建立的模型較精確，他們掌握的解題概念也較完整；建模能力較差的學生建立的模型較模糊，導致他們難以掌握解題的關鍵點。建模是科學領域用來理解現象的重要技能，它能將複雜現象抽象化簡以利思考，使理論容易被理解或進行預測(Gilbert et al., 2000)，透過程式設計建模程序能夠幫助學生們釐清自己對問題的理解，將符合題目的概念進一步抽象化後以找出解題所需的知識概念，再將概念之間聯繫起來，最後完成科學程式設計建模，這種學習方式使學生能對科學程式設計概念加以表徵，降低原本程式設計的學習難度，也使原本抽象的概念變得更容易分析。由此可知，擅長建立程式設計模型的學生擁有較好的問題解決策略，在程式設計的學習上更有成效。

**二、 本研究之學習者透過基於科學運算之運算思維導向程式設計教學後，建模品質較好的學生與建模品質較差的學生在科學學習表現上未有影響**

研究者在本次科學與程式設計課程中進行教學實驗，以與程式設計相同的條件來探討學生們在科學學習的表現，並且在期中考與期末考週分別進行科學前測與後測，將學生依照建模能力高低分組後以ANCOVA分析此數據。依據本次實驗結果發現，學生建模能力好壞並未影響其在科學學習的成效。由於本研究將課程設計重心放在理學院大學生的程式設計學習，科學領域則是輔助學生們進行跨領域學習為目的，且不同科學領域學習的內容差異甚大，本次實驗對象領域背景較分散，科學作為一門需要長期時間去探究的專業科目，在實驗課程時間有限上還須兼顧學生們對課程的需求，因此課程設計無法專精在其中一門科學學科上，以致短期內較難以看出科學能力因建模能力好壞而產生的差異。

**三、 本研究之學習者透過基於科學運算之運算思維導向程式設計教學後，建模品質較好的學生在科學程式設計專題表現上較佳**

研究者在本次科學與程式設計課程中進行教學實驗，並且在實驗課程結束後讓學生進行科學運算專題報告，將學生依照建模能力高低分組後以獨立樣本t檢定分析他們在專題學習中的各項數據。依據本次實驗結果發現，建模能力較好的學生在專題學習上的各項表現都比建模能力較差的學生好。研究者觀察到在專題難易度中，建模能力較好的學生有選擇製作難度較高的科學現象作為題目的趨勢；建模能力較差的學生選擇專題的難度趨勢較不特定。在專題說明書的部分，建模能力較好的學生會用較多的篇幅來介紹科學主題，以及完成專題時的解題思路，此外他們也多半會加入自己對專題的反思與改進專題的方式，報告內容讓人感受到他們經歷的科學探究過程；建模能力較差的學生報告書的內容大多只完成課程所需的最低限度，較難從中看出解題思維。在專題程式碼與專題程式建模的部分，建模能力較好的學生建立的專題模型較完整，他們能夠依據對應主題的科學概念進行詳細的現象描述，找出對應主題的關鍵要素做資料建模，設計符合專題所需的程式解題程序做流程建模，並且能夠依據上述的歷程將想法轉換為程式碼，在整體專題程式的完成度較高；建模能力較差的學生建立的專題模型較淺白且不完整，在第一步驟現象描述中大多是以列點帶過缺少描述的歷程，資料建模時的內容也與現象描述中不符，流程建模中也難以看出解題的程序與邏輯，程式設計往往會有語法或邏輯錯誤導致專題程式不完整。研究者認為，學生在建立程式設計模型時，越清楚的定義模型能夠讓學生更容易將專題模型轉換為程式碼。

**四、 本研究之學習者透過基於科學運算之運算思維導向程式設計教學後，建模品質較好的學生在程式設計建模程序的觀察與除錯中有較多的認同**

研究者在本次科學與程式設計課程中進行教學實驗，並且在期中考與期末考週分別進行科學運算學習態度前測與後測，將學生依照建模能力高低分組後以ANCOVA分析此數據。依據本次實驗結果發現，學生建模能力好壞影響對科學運算學習態度觀察與除錯的看法。研究者發現建模能力較好的學生相較於建模能力較差的學生在建模經驗的觀察與除錯的項目中有較顯著的差異，從研究者在課堂中的觀察，學生們在課堂中開始寫程式前，習慣花較多時間在程式設計平台上建立對應課程主題的程式設計模型，他們會將現象描述、資料建模、流程建模的步驟都完成後，才開始進行程式化的步驟，在課堂觀察中能發現學生們在每次執行程式後會再回頭修改他們的程式設計模型。雖然兩組學生們對於建模程序觀察與除錯的認同降低，但從他們在程式設計學習成效來看，兩組學生的平均成績都有進步，表示程式設計建模程序對他們學習上是有所幫助的。由學生們對程式設計建模程序的看法可知，不論在現象描述、資料建模、流程建模，這些程序都有學生認同其對問題解決的重要性。現象描述是開始寫程式前的第一步驟，能幫助釐清題目的意義進行問題解析依據記錄的想法來了解解題架構後，有助於對科學程式問題有更全面的理解。流程建模的步驟能將想法條列出來，整理出每一步驟該如何進行，可以更清楚解決問題的歷程，也能容易找出需要修改的部分。資料建模的步驟可以將解題所需的參數與變數列出，避免遺漏參數的風險，且能知道每一項目的定義，幫助建立對問題更加完整的思考。整體而言，學生們認為透過科學程式建模程序的引導，能夠讓他們在開始進行科學程式設計前先將想法以文字或圖形的方式列出，方便對問題進行抽象化，且有助於對問題的理解，在向他人解釋程式上也更有脈絡能依循。

在科學程式設計教學時，透過讓學生練習程式建模步驟能引導學生思考科學程式問題，將問題作現象描述、流程及資料建模，完整地釐清自己對該主題的想法後再開始寫程式。學生們普遍認為以科學運算來學習程式設計的課程，有助於學生進行科學與程式設計的科際整合，讓學生以程式設計來嘗試解決真實世界問題，並能幫助他們在程式設計上的學習成效與表現。因此，本次研究為後續CT+X主題的程式設計教學提供了一種可行的方式。

1. 建議

**一、程式設計建模平台**

本實驗透過程式設計建模平台作為輔助學生建立程式設計模型的工具，程式設計建模平台共有5個步驟：現象描述、資料建模、流程建模、程式化以及觀察與除錯，雖然學生們普遍都認同建模對他們學習程式帶來的助益，但也有些學生認為平台上的功能能夠再改善。例如：資料建模的限制較多，在使用上有時會不夠直覺而容易產生困難。程式化的部分則是希望能夠改進程直接在平台上練習寫程式，而不需使用其他的Python編譯器。

**二、科學程式設計課程**

本次科學與程式設計課程中，我們對理學院大學生進行建模導向的科學程式設計教學，嘗試以建模步驟引導學生探究科學現象並學習程式設計，其成果顯示此種以建模為主的實驗教學方式有助於學生在程式設計上的學習，雖然未看出對科學的學習成效，但仍然有其價值所在。研究者將課程重心大多放在程式設計教學上，科學則是作為提供真實情境問題使學生從中體會程式設計的輔助內容。儘管，建模能力好壞對學生在科學表現上並未有差異，但仍可觀察到建模能力較好的學生在專題製作上大多會選擇較為複雜的科學主題，展現了他們對於科學探究的興趣。另外，本次課程中使用之科學程式設計建模平台有五個步驟，然而在實驗中學生大多容易忽略最後一項觀察與除錯的步驟，其原因來自課程時間僅每週2節課，每節課50分鐘，但是探究科學現象以及解題思維的培養需要較長且完整的時間才能掌握。因此建議未來在相關實驗中，能調整課程進行的時序，讓學生有更多探究科學與程式設計的時間，多給予學生透過程式探索科學的機會，使STEM教育能夠更加展現其效用。

參考文獻

**英文部分**

Abu-Naser, S. S. (2008). Developing visualization tool for teaching AI searching algorithms.

Alessi, S. M., & Trollip, S. R. (2001). Multimedia for learning: Methods and development. Allyn & Bacon.

Barella, A., Valero, S., & Carrascosa, C. (2008). JGOMAS: New approach to AI teaching. IEEE Transactions on education, 52(2), 228-235.

Bellstrom, P., Thoren, C. (2009). Learning how to program through visualization: A pilot study on the bubble sort algorithm. 2009 Second International Conference on the Applications of Digital Information and Web Technologies (Icadiwt 2009), 90–94. doi:10.1109/icadiwt.2009.5273943

Burgsteiner, H., Kandlhofer, M., & Steinbauer, G. (2016, March). Irobot: Teaching the basics of artificial intelligence in high schools. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1).

Chen, Y. L., Hong, Y. R., Sung, Y. T., & Chang, K. E. (2011). Efficacy of simulation-based learning of electronics using visualization and manipulation. Journal of Educational Technology & Society, 14(2), 269-277.

Colaso, V., Kamal, A., Saraiya, P., North, C., McCrickard, S., & Shaffer, C. (2002). Learning and retention in data structures: A comparison of visualization, text, and combined methods. Paper presented at the Proceedings of ED-MEDIA 2002, June 24-29, Denver, Colorado, USA.

Cuéllar, M. P., & Pegalajar, M. C. (2014). Design and implementation of intelligent systems with LEGO Mindstorms for undergraduate computer engineers. Computer Applications in Engineering Education, 22(1), 153-166.

De Jong, T., & Van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. Review of educational research, 68(2), 179-201.

Diehl, S. (2007). Software visualization: visualizing the structure, behaviour, and evolution of software. Springer Science & Business Media.

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world. British Journal of Educational Technology, 42(4), 624-637.

Estevez, J., Garate, G., Guede, J. M., & Grana, M. (2019). Using Scratch to Teach Undergraduate Students' Skills on Artificial Intelligence. arXiv preprint arXiv:1904.00296.

Fernandes, M. A. (2016). Problem‐based learning applied to the artificial intelligence course. Computer Applications in Engineering Education, 24(3), 388-399.

Garay, G. R., Tchernykh, A., Drozdov, A. Y., Garichev, S. N., Nesmachnow, S., & Torres-Martinez, M. (2017). Visualization of VHDL-based simulations as a pedagogical tool for supporting computer science education. Journal of Computational Science.

Grivokostopoulou, F., Perikos, I., & Hatzilygeroudis, I. (2014, December). Using semantic web technologies in a web based system for personalized learning AI course. In 2014 IEEE Sixth International Conference on Technology for Education (pp. 257-260). IEEE.

Gurka, J. S., & Citrin, W. (1996, September). Testing effectiveness of algorithm animation. In Proceedings 1996 IEEE Symposium on Visual Languages (pp. 182-189). IEEE.

Hansen, S. R., Narayanan, N. H., & Schrimpsher, D. (2000). Helping learners visualize and comprehend algorithms. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 2(1), 10.

Homer, B. D., & Plass, J. L. (2014). Level of interactivity and executive functions as predictors of learning in computer-based chemistry simulations. Computers in Human Behavior, 36, 365-375.

Hošková-Mayerová, Š., & Rosická, Z. (2015). E-learning pros and cons: active learning culture?. Procedia-Social and Behavioral Sciences, 191, 958-962.

Hundhausen, C., & Douglas, S. (2000, September). Using visualizations to learn algorithms: should students construct their own, or view an expert's?. In Proceeding 2000 IEEE International Symposium on Visual Languages (pp. 21-28). IEEE.

Hundhausen, C. D., Douglas, S. A., & Stasko, J. T. (2002). A meta-study of algorithm visualization effectiveness. Journal of Visual Languages & Computing, 13(3), 259-290.

Huppert, J., Yaakobi, J., & Lazarowitz, R. (1998). Learning microbiology with computer simulations: Students’ academic achievement by method and gender. Research in Science & Technological Education, 16(2), 231-245.

Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. Journal of Computer Science Integration, 1(1).

Jarc, D. J., Feldman, M. B., & Heller, R. S. (2000). Assessing the benefits of interactive prediction using web-based algorithm animation courseware. ACM SIGCSE Bulletin, 32(1), 377-381.

Jensen, D., Self, B., Rhymer, D., Wood, J., & Bowe, M. (2002). A rocky journey toward effective assessment of visualization modules for learning enhancement in Engineering Mechanics. Journal of Educational Technology & Society, 5(3), 150-162.

Jonassen, D. H., & Strobel, J. (2006). Modeling for meaningful learning. In Engaged learning with emerging technologies (pp. 1-27). Springer, Dordrecht.

Kandlhofer, M., Steinbauer, G., Hirschmugl-Gaisch, S., & Huber, P. (2016, October). Artificial intelligence and computer science in education: From kindergarten to university. In 2016 IEEE Frontiers in Education Conference (FIE) (pp. 1-9). IEEE.

Kochlán, M., & Hodon, M. (2014, September). Open hardware modular educational robotic platform—Yrobot. In 2014 23rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD) (pp. 1-6). IEEE.

Korhonen, A., & Malmi, L. (2000). Algorithm Simulation with Automatic Assessment. Paper presented at the 5th Annual ACM SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2000). Helsinki, Finland.

Krishnan, D. G., Keloth, A. V., & Ubedulla, S. (2017). Pros and cons of simulation in medical education: A review. Education, 5, 7.

Kumar, A. N. (2004). Three years of using robots in an artificial intelligence course: lessons learned. Journal on Educational Resources in Computing (JERIC), 4(3), 2-es.

Lawrence, A. W., Badre, A. M., & Stasko, J. T. (1994, October). Empirically evaluating the use of animations to teach algorithms. In Proceedings of 1994 IEEE Symposium on Visual Languages (pp. 48-54). IEEE.

Marković, M., Kostić Kovačević, I., Nikolić, O., & Nikolić, B. (2015). INSOS—educational system for teaching intelligent systems. Computer Applications in Engineering Education, 23(2), 268-276.

McNally, M., Naps, T., Furcy, D., Grissom, S., & Trefftz, C. (2007). Supporting the rapid development of pedagogically effective algorithm visualizations. Journal of Computing Sciences in Colleges, 23(1), 80-90.

Mintz, R. (1993). Computerized simulation as an inquiry tool. Scool Science and Mathematics, 93(2), 76-80.

Moyer-Packenham, P. S., Lommatsch, C. W., Litster, K., Ashby, J., Bullock, E. K., Roxburgh, A. L., ... & Clarke-Midura, J. (2019). How design features in digital math games support learning and mathematics connections. Computers in Human Behavior, 91, 316-332.

Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., et al. (2003). Exploring the role of visualization and engagement in computer science education. ACM SIGCSE Bulletin, 35(2), 131-152.

O'Neil, H. F., Wainess, R., & Baker, E. L. (2005). Classification of learning outcomes: Evidence from the computer games literature. The Cirriculum Journal, 16(4), 455-474.

Pedro, F., Subosa, M., Rivas, A., & Valverde, P. (2019). Artificial intelligence in education: Challenges and opportunities for sustainable development.

Prensky, M. (2002). The Motivation of Gameplay or, the REAL 21th century learning revolution. URL http://www. marcprensky. com/writing/Prensky, 2010-1.

Russell, S., & Norvig, P. (2002). Artificial intelligence: a modern approach.

Saraiya, P., Shaffer, C. A., McCrickard, D. S., & North, C. (2004, March). Effective features of algorithm visualizations. In Proceedings of the 35th SIGCSE technical symposium on Computer Science Education (pp. 382-386).

Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cunniff, D., ... & Verno, A. (2011). CSTA K--12 Computer Science Standards: Revised 2011. ACM.

Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition.

Shaffer, C. A., Cooper, M. L., Alon, A. J. D., Akbar, M., Stewart, M., Ponce, S., & Edwards, S. H. (2010). Algorithm visualization: The state of the field. ACM Transactions on Computing Education (TOCE), 10(3), 1-22.

Simoňák, S. (2016, January). Algorithm visualizations as a way of increasing the quality in computer science education. In 2016 IEEE 14th international symposium on applied machine intelligence and informatics (SAMI) (pp. 153-157). IEEE.

Sklar, E., Eguchi, A., & Johnson, J. (2002, June). RoboCupJunior: learning with educational robotics. In Robot Soccer World Cup (pp. 238-253). Springer, Berlin, Heidelberg.

Stone, P., Brooks, R., Brynjolfsson, E., Calo, R., Etzioni, O., Hager, G., ... & Teller, A. (2016). Artificial intelligence and life in 2030: the one hundred year study on artificial intelligence.

Thomas, R., & Neilson, I. (1995). Harnessing simulations in the service of education: The interact simulation environment. Computers & Education, 25(1-2), 21-29.

Tudoreanu, M. E., Wu, R., Hamilton-Taylor, A., & Kraemer, E. (2002, September). Empirical evidence that algorithm animation promotes understanding of distributed algorithms. In Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments (pp. 236-243). IEEE.

Tversky, B., Morrison, J. B., & Betrancourt, M. (2002). Animation: can it facilitate?. International journal of human-computer studies, 57(4), 247-262.

Végh, L., & Stoffová, V. (2017). Algorithm animations for teaching and learning the main ideas of basic sortings. Informatics in Education, 16(1), 121-140.

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

Yoon, D. M., & Kim, K. J. (2015). Challenges and opportunities in game artificial intelligence education using Angry Birds. IEEE Access, 3, 793-804.

**中文部分**

教育部（2018），十二年國民基本教育課程綱要 國民中學暨普通型高級中等學校 科技領域，教育部。

黃仁暐、涂益郎（2019），和AI做朋友-相知篇，教育部。

附錄一 程式設計能力測驗

**一、程式設計能力測驗前測題目**

|  |  |
| --- | --- |
| 基本題 | 1. 民國與西元  已知今年是民國108年，也是西元2019年，請寫一個程式能讀入一個正整數 N 代表民國年，並輸出其對應的西元年。 |
| **2. 時間換算I**  已知一小時是60分鐘，假設現在有 h 小時和 m 分鐘，請問這樣總共是幾分鐘？ 輸入為兩個整數 H、M 代表有 H 小時和 M 分鐘，請輸出總共是幾分鐘。 |
| **3. 時間換算II**  已知一小時是60分鐘，假設現在有 m 分鐘，請問你它總共是幾小時、幾分鐘？ 輸入一個整數 M 代表有 M 分鐘，輸出總共是幾小時、幾分鐘。 |
| **4. 比大小**  請寫一個程式讀入兩個相異整數a, b，並輸出較小的那一個數。 |
| **5. 幾科及格?**  已知60分為及格，請寫一個程式讀入10個科目成績，並輸出幾科及格。 |
| **6 九九乘法表**  請寫一個程式幫忙印出單項的九九乘法表。輸入是一個數字N，輸出為N的乘法表。 |
| **7. 數字ABC**  有三個正整數A、B 和C。若依序輸入A+B、A+C、B+C的值，那麼A=? B=? C=?例如：A+B=16，A+C=20，B+C=18，則A=9，B=7，C=11。但如果找不出整數的A，B，C符合條件，則輸出"No solution." |
| **8. 3的餘數**  請寫一個程式讀入一個正整數N，接下來讀入N個正整數，並依序輸出有幾個數被3除餘0?被3除餘1? 被3除餘2? |
| 進階題 | **9. 生動的combo hits**  請寫一個程式，讀入一個正整數N，先輸出一句"Fight!"接著顯示生動的N combo hits。所謂生動是指combo hits會以下面的方式陳列。 |
| **10. 規則數列**  有一個規則數列前10項如下： 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55,... 請觀察此數列的規則，寫一個程式讀入一個正整數N，印出此數列的前N項。 |
| **11. 123數列**  假設我們定義一個「123數列」，它的第0項是1、第1項是2、第2項是3，接下來的每一項都是前三項的和。也就是說，這數列的前幾項為：1, 2, 3, 6, 11, 20, 37, 68, 125, 230 ...。請寫一個程式讀入一個整數 N ，並輸出此數列的第 N 項。 |
| **12. 字謎ABCD**  請找出所有的(A, B, C, D)組合，使得A≠B ≠C ≠D且以下算式成立 AB + BC = CD 輸出時請依照A，B，C，D的字典順序輸出，也就是先輸出A較小的，當A相同時，依序比較B、C、D，較小的先輸出。 |

**二、程式設計能力測驗後測題目**

|  |  |
| --- | --- |
| 1 | 有一個規則數列前10項如下： 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55,... 請觀察此數列的規則，寫一個程式讀入一個正整數N，印出此數列的前N項。 |
| 2 | 請寫一個程式讀入一個正整數N，接下來讀入N個正整數，並依序輸出有幾個數被4除餘0？被4除餘1？ 被4除餘2？ 被4除餘3？ |
| 3 | 請說明下列程式目的為何？若print(X(36, 8))會得到多少？ |
| 4 | 請將下列的程式碼改寫成遞迴的結構。 |

附錄二 科學能力測驗

**一、科學能力測驗前測題目**

|  |  |
| --- | --- |
| 1 | 自高塔頂端靜止落下一鐵塊。不考慮空氣阻力，如果此一鐵塊是自由落體，經4秒鐘後著地，試問塔高多少公尺？(g＝9.8 )。(應用題，請寫下計算過程) |
| 2 | 根據理想氣體方程式(Ideal Gas Law)，當容器內的體積(V)變小時，壓力(P)會怎麼變化？為什麼？ |
| 3 | 我們知道生物的DNA是由A、C、G、T所組成的片段，假設有四種生物的DNA片段如下： A: A C G G T A C C C T T A T G C A A T A B: C T G A C G G C C A C A T T C A T C C: G C A C G C C C C G T A C G T A C A D: C A C G C C C G G T A A T T A T G C A 請問生物A的DNA片段和B,C,D誰比較相近？為什麼？ |
| 4 | 請以數學式來表示以下數列的遞迴關係(Recursion)： 1 1 3 7 17 41 99 239 577 1393 |

**二、科學能力測驗後測題目**

|  |  |
| --- | --- |
| 1 | 英國經濟學家馬爾薩斯，在1798年提出「人口的成長率與總人口數成正比。」的人口成長模型。令 P(t) 表時間 t 的人口數，且 P0 為 t = t0 時的人口數，則以上敘述可以數量化為:  \begin{displaymath}\left\{ \begin{array}{l} P'(t) =\lambda P(t), \quad \lambda >0 \\ P(t_0) = P_0 >0 \end{array} \right. \end{displaymath} 請問上述模型對於真實人口成長忽略了哪些因素(請寫出至少三項)? |
| 2 | 若以初速度v1鉛直上拋一小球，經過t1秒上升至最高點，再經過t2秒小球落回初拋點，速度變為v2，若空氣阻力的影響不能忽略，請比較v1 v2的大小、比較t1 t2的大小，並說明為什麼? |
| 3 | 我們知道生物的DNA是由A、C、G、T所組成的片段，假設有四種生物的DNA片段如下： A: A C G G T A C C C T T A T G C A A T A B: C T G A C G G C C A C A T T C A T C C: G C A C G C C C C G T A C G T A C A D: C A C G C C C G G T A A T T A T G C A 請問生物A的DNA片段和B,C,D誰比較相近？為什麼？ |
| 4 | 請以數學式來表示以下數列的遞迴關係(Recursion)： 1 1 3 7 17 41 99 239 577 1393 |

附錄三 科學程式設計態度問卷

**一、前測態度問卷題目**

|  |  |  |  |
| --- | --- | --- | --- |
| 面向 | 類別 | 編號 | 題目 |
| 建模經驗 | 現象描述 | 1 | 遇到程式或科學問題時，我通常能夠很快理解問題並大致描述可能的解題方法。 |
| 2 | 我認為現象描述的歷程能幫助我做問題解析，使我更能掌握解題的關鍵與規則。 |
| 流程建模 | 3 | 我覺得在解程式或科學問題時，先思考嚴謹的解題流程對解題是很有幫助的。 |
| 4 | 我認為流程建模能幫助我在寫程式時有更清晰的解題邏輯。 |
| 資料建模 | 5 | 我在面對科學或程式問題時，會嘗試分析問題中的各種變因，以幫助問題解決。 |
| 6 | 我認為資料建模能幫助我思考問題的變因該如何在程式中定義與表示，以及各個變因間的關聯性。 |
| 程式化 | 7 | 每次思考完解題的方法，轉換成程式碼時仍會遇到許多問題。 |
| 8 | 透過現象描述、資料建模及流程建模的步驟後，我通常能順利寫出程式。 |
| 觀察 與除錯 | 9 | 課堂裡觀察與除錯的步驟中，讓我有機會思考自己程式邏輯或語法上的錯誤。 |
| 10 | 找到程式中的錯誤是寫程式很困難的部分。 |
| 自我效能 | 程式設計 | 11 | 每當我遇到程式設計問題時，我總是很難聯想到題目所需的程式概念，不知道寫程式第一步該如何進行。 |
| 12 | 即使是複雜的程式問題，我也能夠透過觀察題意中的脈絡，逐步思考問題的解法。 |
| 13 | 即使我在程式設計上遇到困難，我仍然會持續努力分析自己程式的問題直到問題解決。 |
| 14 | 我寫程式時通常是獨力完成，不需要藉由同儕或老師的幫助。 |
| 科學 | 15 | 每當我遇到科學問題時，我總是很難聯想到題目所需的科學概念，不知道如何開始探索與解決科學問題。 |
| 16 | 面對複雜的科學問題時，我也能透過分析問題的脈絡，以進行可能的假設。 |
| 17 | 我通常獨力解決科學問題，不需藉由同學或老師的幫助。 |
| 18 | 即使看著科學定義或公式，我仍無法理解其背後的科學概念與涵義。 |
| 學習態度 | 程式設計 | 19 | 我認為學習程式設計能對我未來的職業有幫助。 |
| 20 | 我認為日常生活中有許多地方會用到程式設計的技能。 |
| 科學 | 21 | 我認為學習科學能對我未來的職業有幫助。 |
| 22 | 我認為日常生活中有許多地方都會用到科學的概念或原理。 |
| 科學 程式設計 | 23 | 藉著寫程式解決科學問題，讓我更加理解程式設計的邏輯與方法。 |
| 24 | 寫程式解科學問題的過程幫助我更深入理解科學現象。 |
| 學習興趣 | 程式設計 | 25 | 我認為學習程式設計是有趣的。 |
| 26 | 我願意主動學習程式設計。 |
| 科學 | 27 | 我認為學習科學是有趣的。 |
| 28 | 我願意主動學習科學相關議題。 |

**二、後測態度問卷題目**

|  |  |  |  |
| --- | --- | --- | --- |
| 面向 | 類別 | 編號 | 題目 |
| 建模經驗 | 現象描述 | 1 | 遇到程式或科學問題時，我通常能夠很快理解問題並大致描述可能的解題方法。 |
| 2 | 我認為現象描述的歷程能幫助我做問題解析，使我更能掌握解題的關鍵與規則。 |
| 流程建模 | 3 | 我覺得在解程式或科學問題時，先思考嚴謹的解題流程對解題是很有幫助的。 |
| 4 | 我認為流程建模能幫助我在寫程式時有更清晰的解題邏輯。 |
| 資料建模 | 5 | 我在面對科學或程式問題時，會嘗試分析問題中的各種變因，以幫助問題解決。 |
| 6 | 我認為資料建模能幫助我思考問題的變因該如何在程式中定義與表示，以及各個變因間的關聯性。 |
| 程式化 | 7 | 每次思考完解題的方法，轉換成程式碼時仍會遇到許多問題。 |
| 8 | 透過現象描述、資料建模及流程建模的步驟後，我通常能順利寫出程式。 |
| 觀察 與除錯 | 9 | 課堂裡觀察與除錯的步驟中，讓我有機會思考自己程式邏輯或語法上的錯誤。 |
| 10 | 找到程式中的錯誤是寫程式很困難的部分。 |
| 自我效能 | 程式設計 | 11 | 每當我遇到程式設計問題時，我總是很難聯想到題目所需的程式概念，不知道寫程式第一步該如何進行。 |
| 12 | 即使是複雜的程式問題，我也能夠透過觀察題意中的脈絡，逐步思考問題的解法。 |
| 13 | 即使我在程式設計上遇到困難，我仍然會持續努力分析自己程式的問題直到問題解決。 |
| 14 | 我寫程式時通常是獨力完成，不需要藉由同儕或老師的幫助。 |
| 科學 | 15 | 每當我遇到科學問題時，我總是很難聯想到題目所需的科學概念，不知道如何開始探索與解決科學問題。 |
| 16 | 面對複雜的科學問題時，我也能透過分析問題的脈絡，以進行可能的假設。 |
| 17 | 我通常獨力解決科學問題，不需藉由同學或老師的幫助。 |
| 18 | 即使看著科學定義或公式，我仍無法理解其背後的科學概念與涵義。 |
| 學習態度 | 程式設計 | 19 | 我認為學習程式設計能對我未來的職業有幫助。 |
| 20 | 我認為日常生活中有許多地方會用到程式設計的技能。 |
| 科學 | 21 | 我認為學習科學能對我未來的職業有幫助。 |
| 22 | 我認為日常生活中有許多地方都會用到科學的概念或原理。 |
| 科學程式設計 | 23 | 藉著寫程式解決科學問題，讓我更加理解程式設計的邏輯與方法。 |
| 24 | 寫程式解科學問題的過程幫助我更深入理解科學現象。 |
| 學習興趣 | 程式設計 | 25 | 我認為學習程式設計是有趣的。 |
| 26 | 我願意主動學習程式設計。 |
| 科學 | 27 | 我認為學習科學是有趣的。 |
| 28 | 我願意主動學習科學相關議題。 |
| 學期課程 | 開放式 | 29 | 請問在這學期的課程中，透過解決科學問題的方式來學程式設計，是否讓你更容易學程式設計？為什麼？ |
| 30 | 請問科學程式設計平台中的哪些步驟對你學程式設計是有用的？為什麼？ |