

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

計畫編號/Project Number：PEE1090711

學門專案分類/Division：工程

執行期間/Funding Period：2020-08-01-2022-01-31

非資訊相關科系學生在混成式與線上程式設計課程中學習成效之比較研究
初級程式設計-Python

計畫主持人(Principal Investigator)：賴志宏

共同主持人(Co-Principal Investigator)：無

執行機構及系所(Institution/Department/Program)：

國立東華大學 資訊工程學系

延後公開

繳交報告日期(Report Submission Date)：2022 年 3 月 20 日

非資訊相關科系學生在混成式與線上程式設計課程中學習成效之比較研究

摘要

我國教育部要求至少半數大學生畢業前修過程式設計，不過短期內要在校內針對非資訊科系的學生大量開課，仍面臨師資、與設備不足的挑戰，若以線上課程進行，其具有不受時空限制的優點，是否可解決這兩項問題。另外，線上課程缺乏面對面的互動，若是採用實體與線上結合的混成學習，是否能兼具線上與實體的優點，讓程式設計的學習達到更好的效果。其次，對於非資訊相關科系的學生，其資訊相關知識與背景較少，需要不同的學習內容與學習方式，以及提供更多學習上的輔助。因此本計畫的目的有兩部分，首先針對非資訊相關科系學生的需求，開發了一套程式學習系統與數位教材。並進一步探討非資訊科系學生在純線上與混成式課程對學習的影響。研究對象主要為大學部一年級非資訊相關科系的學生，在兩個基礎程式設計的選修課程中，分別採取純線上、與混成的方式進行。研究結果顯示兩個班級學生在學習動機、自我效能、學習滿意度、學習成效上並無顯著差異。此外，本研究也針對學生的回饋進行分析，並提供教學與研究上的建議。

關鍵詞: 程式設計學習、非資訊相關科系學生、混成式課程、線上課程

Comparison of learning effect between blended and online programming courses on non-information related students

Abstract

The Ministry of Education of our country requires at least half of the college students to take a computer programming course before graduation. However, it has challenges on insufficient teachers and equipment to provide a large number of courses for non-information-related students in the short term. Online courses can be conducted without limitations of time and space. Whether the advantages of online courses can solve these two problems. In addition, online courses lack face-to-face interaction. Whether blended courses which combine physical and online courses can improve programming learning? Secondly, non-information-related students who have less knowledge and background of programming need different learning content, learning methods, and more learning assistance. Therefore, this project has two purposes. First, it is to develop a program learning system and digital teaching materials to meet the needs of those students to learn computer programming. Second, the project also further explored the impact of online and blended courses on programming learning. The participants are mainly first-year non-information-related students who enrolled in the two elective courses of basic programming in which online and blended methods are used respectively. The results of the study showed that there was no significant difference in learning motivation, self-efficacy, learning satisfaction, and learning achievement between the two classes. In addition, this study also analyzes student feedback and provides teaching and research recommendations.

Keyword: programming learning, non-information related students, blended course, online course

目 錄

一、 研究動機與目的.....	1
1. 教學實踐研究計畫動機.....	1
2. 研究目的。.....	2
二、 文獻探討.....	2
1. 非資訊相關科系學生之程式設計教學相關研究.....	2
2. 線上與混成式課程對程式設計學習的影響.....	4
三、 研究問題.....	6
四、 研究設計與方法.....	6
1. 研究架構與對象.....	6
2. 程式設計學習系統.....	7
3. 研究工具.....	8
五、 教學暨研究成果.....	9
1. 教學過程與成果.....	9
2. 教師教學反思.....	9
3. 學生學習回饋.....	9
六、 建議與省思.....	10
參考文獻.....	11

非資訊相關科系學生在混成式與線上程式設計課程中學習成效之比較研究

一、研究動機與目的

1. 教學實踐研究計畫動機

程式設計不但是資訊素養必備的知識與技能，也是發展資訊科技的重要工具，近年來全球掀起一股程式設計的教育浪潮，包含愛沙尼亞、英國、法國、奧地利、丹麥、及波蘭等各國，均將程式語言教學納入課程綱要中（王令宜，2017），我國教育部也在2019年實施的十二年國教的課綱中，增列了科技領域，並將程式設計等資訊科技內容列入國、高中的必修課程中。除此之外，在高教深耕計畫中，也單獨列出一項指標，要求我國大學不分學系，五年內至少要有半數學生，在畢業前修過程式設計的課程（聯合報，2018），可見世界各先進國家對程式設計課程的重視。本校為培育人才也不遺餘力，訂定了程式設計課程的實施辦法，要求所有新進的學士班學生需修習程式設計課程，或通過程式能力檢定，修習程式設計課程的人數，因此而暴增了約2,000人。龐大的修課需求同時也帶來了許多的考驗，例如要同時增加可能數千名的學生使用電腦教室，學校是否能夠負荷；對於資訊教師也將增加龐大的課務負擔；此外，現有的程式設計課程大部分是針對資訊相關科系的學生設計，其教學內容是否適合非資訊相關科系的學生學習，也會是另一個值得探討的議題。

近幾年，拜網際網路快速發展與教育政策的改變，線上課程大量在高等教育中崛起（何榮桂，2014），尤其是在Coursera、Udacity、edX和Khan Academic等平台開設了大規模網路開放課程（Massive Open Online Course, MOOC，或稱為磨課師）之後，國內、外許多大學紛紛設置類似平台，以開設線上課程。磨課師採用大班且線上的上課方式，可以打破學習時間與學習空間的限制（鄭念慈，2017），減少電腦教師與電腦教室的需求數量。因此，若全面推行類似的線上課程，來進行程式設計的教學，似乎可以改善上述設備與師資不足的問題。本校為了因應全面開設的程式設計課程，除了實體課程之外，也開設了許多線上課程，以滿足這龐大的修課需求。申請人在最近3個學期皆開設了線上課程，已修課人數已經累積超過800人。但是課程中面臨了許多線上課程常遇到的困難，例如缺乏傳統課程中面對面的互動，以及時空的分隔常造成學習者的學習孤立感，因而當學生遇到問題或學習瓶頸時，可能因無法得到立即的協助而中輟了學習（王梅玲，2017），造成低完課率（complete rate）與學習成效不高的問題（劉怡甫，2014）。為改善純線上課程所遭遇的限制，有許多學者提倡混成式課程（blended course）。混成式課程的優點在於它結合了實體與網路等科技的學習，因此被廣泛應用在各種不同的領域（陳芷如，2017），Güzer與Caner（2014）指出學生喜歡網路的學習環境，但是並不想放棄面對面的學習機會，這顯示學習者認為網路和實體兩種學習環境都各有優點，應該同時具備。另外，史美瑤（2014）綜合文獻後指出混成學習有以下優點：1. 學生的學習成效較高；2. 增加學生與學生之間以及與老師的互動；3. 促使教師改變教學方法和教學設計；與4. 保留學生學習的過程與成果。有鑑於此，本計畫希望採用混成式課程，一方面減少校內電腦設備的使用需求，並透過經過專門培訓的助教的協助，來進行程式設計的教學，希望能結合網路科技與面對面互動兩者的優勢，以解決師資與設備不足的問題，並同時能增進學生學習的成效。

另外，程式設計課程在資訊相關科系的課程中，常常被許多學生認為是相當困難的一個科目（Tan, Guo, Zheng, & Zhong, 2014），學生在學習上容易遇到許多困難（Urquiza-Fuentes & Velazquez-Iturbide, 2013），因而造成課程中輟率的居高不下（Denny, Luxton-Reilly, Tempero, & Hendrickx, 2011）。對於資訊相關科系的學生尚且不易學習，對於非資訊相關科系的學生而言，由於他們缺乏相關的基礎知識、相關課程的支持，而且學生不是他們的主修，又是在被要求而非自願來修課的情況下，主動學習意願的不足，可能造成學習上更大的困難（Dawson, Allen, Campbell, & Valair, 2018; Sattar & Lorenzen,

2009)，因此對非資訊相關科系學生的課程，有必要提供更多的學習輔助。尤其是在線上或是混成式課程中，學生絕大部分的學習活動都是透過線上學習系統來完成，因此學習系統扮演著成敗關鍵的重要角色，例如自動但保有彈性的程式批改功能，以及有良好的同儕互動機制等，都會影響學生的學習成效。因此本研究也將開發一套適合非資訊相關科系學生使用的的程式設計學習系統。

2. 研究目的。

針對上述動機，本研究計畫之目的為：

- (1) 針對非資訊相關科系學生的學習困難，建置一套適合其學習的程式設計學習系統；
- (2) 比較非相關科系學生在純線上與混成式之程式設計課程中，其學習動機、學習滿意度、自我效能、以及學習成效的差異。

二、文獻探討

1. 非資訊相關科系學生之程式設計教學相關研究

蘋果公司的執行長庫克說：學程式語言比學英語重要（遠見雜誌，2017），可見程式設計不只是專業軟體工程師才需要，它可以訓練邏輯思考，學習從不同角度看待問題，提高面對生活挑戰的能力，為想要解決的問題找到一種途徑，是未來人人必須要會的新技能（數位時代，2018）。程式設計的目的，是希望將問題的解法透過我們撰寫的程式碼，指揮電腦依照我們設定好的流程逐步去執行，以完成特定的任務。一般開發程式的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，若有錯誤，則必須除錯，直到完全修正錯誤為止(Deek, Kimmel, & McHugh, 1998)。

因此程式設計是一個複雜的過程，在進行程式設計的問題解決流程中，電腦內部執行的狀況以及儲存的資料內容是我們無法看到或接觸的（Shih & Alessi, 1993），因此要學好程式設計課程必須具備多種能力或學習過程，除了要能記憶複雜的敘述與語法（Rodrigo et al., 2013）之外，還要具備抽象概念的理解（Lahtinen, Ala-Mutka & Jaärvinen, 2005）、問題解決（Gomes & Mendes, 2007）、以及後設認知（陳婉寧，2011）的能力，使得新手的學習是困難的（Zhang, Liu, de Pablos, & She, 2014），以下列出程式設計教學時，學生學習常見的困難點（陳婉寧，2011）：

- (1) 複雜的指令與語法的記憶困難：除了 Alice 或是 Scratch 等圖形化的程式語言之外，一般的程式語言需要記憶程式的許多敘述或是指令，對程式設計者是一項負荷，尤其英文程度不好的學生更是雪上加霜。但若專注於在繁瑣的語法規則上，也容易阻礙學習者學的問題解決，導致其不易從整體的角度去思考問題（McIver, 2000）。
- (2) 抽象概念的理解障礙：學習程式設計時對抽象的概念要有正確的理解，但傳統程式設計的教學方式，通常是由教師口頭講解程式概念，然而這些程式概念與執行過程往往無法看見，只憑口語難以具體描述，特別是初學者本身先備知識不足，沒有足夠的具體經驗，便容易對抽象概念有理解上的障礙（Lahtinen, Ala-Mutka, & Jaärvinen, 2005）。
- (3) 缺乏轉化演算法為程式碼的能力：有些程式設計學習者雖然能夠理解演算過程，但卻仍不清楚如何將這些敘述進行有意義的結合，把規劃好的問題解決方案轉化成為程式碼（Winslow, 1996）。
- (4) 學習回饋的不足：在程式設計學習時，學生需要獲得建設性與矯正性回饋（Bancroft & Roe, 2006），以瞭解自己的程式有哪些缺失、應如何修正。然而在實際課堂上，往往為顧及課

程進度，學習者缺乏足夠的資源和適性的指導 (Lahtinen, Ala-Mutka & Järvinen, 2005)，教師亦無法針對每一位學習者不同的學習情況一一給予回饋 (Gomes & Mendes, 2007)

(5) 缺乏鷹架：程式設計時，學生通常需要將程式由無到有，產生完整的程式，這對許多學生是困難的任務，需要外多的鷹架輔助方能使學習更有效 (Chrysafiadi & Virvou, 2013)。

(6) 練習的不足：練習對學習程式設計非常重要 (Leiva & Salas, 2013)，但課堂的練習時間不足，且學生對老師出的作業是否答對，未能有一即時的回饋，降低練習的意願，造成學習的困難 (Liu, 2013)。

總之，程式設計是高度複雜的思考歷程，需要密集的邏輯思考與良好的問題解決能力 (陳婉寧, 2015)，因此，對於資訊相關與非相關科系的學生都是不容易的，不過對資訊相關科系的學生而言，因為這是自己的主要科目，通常有較高的學習動機，但是對非資訊相關科系的學生來說，若是在被要求的情況下修習這門課程，不但學習動機容易低落，完課率也會比較低 (Dawson, Allen, A., & A., 2018; Tek, Benli, & Deveci, 2018)。另外，由於他們缺乏相關的基礎知識、相關課程的支持，造成學習上更大的困難 (Dawson et al., 2018; Sattar & Lorenzen, 2009)。

有鑒於上述程式設計教學的困難，國內外學者提出了許多解決的辦法，例如為了減少指令或敘述的語法的背誦問題，許多研究使用 Alice (陳婉寧, 2011; Chang, 2014; Zhang et al., 2014;) 或 Scratch (梁紫筠, 2015; 劉正吉, 2011) 等視覺化程式語言進行教學，這些軟體已將指令或敘述用圖形的方式呈現，程式設計者不需要背誦指令，只要拖拉圖形化的指令移到適當的流程位置即可；另外，有許多研究採用視覺化的軟體模擬出程式執行時電腦內部資料的改變過程 (Chou & Sun, 2013; Sorva, Karavirta, & Malmi, 2013)，使學習者能瞭解抽象的執行過程。另外有學者使用案例式教學 (Tan et al., 2014) 增進對程式的瞭解，另外，在同儕互動的教學方式方面，同儕教學 (呂國泰, 2013; 郭士豪, 2011)、學習社群 (陳柏翰, 2014)，合作學習 (康錦程, 2009; 蕭俊卿, 2009; 蘇國源, 2013) 也被用在程式設計的教學。為了增加練習的機會，online judge 等程式自動評分系統 (吳承璋, 2014; Liu, 2013) 也被提出，以及遊戲式學習的方式輔助程式設計課程 (Corral, Balcells, Estevez, Moreno, & Ramos, 2014; Gonzalez et al., 2013)，還有使用透過操作機器人來學習程式設計 (Kazakoff & Bers, 2014)，以及使用智慧型教學系統來進行教學 (Jurado, Redondo, & Ortega, 2014)。

然而，至今對於大學程式設計教學的研究，大部分的研究對象都是資訊相關科系的學生 (例如 Garcia-Beltran, Tapia, & Sanchez-Naranjo, 2014; Martinez, Licea, Juarez, & Aguilar, 2014; 吳承璋、孔崇旭, 2014; 許逸揚, 2017)，而資訊相關與非相關之學生的學習動機、先備知識與學習模式不一樣 (Forte & Guzdial, 2005)，過去針對資訊相關科系學生的研究結果，是否能應用或推論到非資訊相關科系的學生，仍有待進一步證實。雖然近年有少部份研究針對非資訊相關科系學生進行研究，例如 Tek 等人 (2018) 對 193 位土耳其非資訊相關科系的學生，分析自我效能對程式設計學習的影響；另外，Echeverria、Cobos、Machuca 與 Claros (2017) 探討合作學習對非資訊相關科系學生的程式設計學習的影響；Xinogalos (2016) 以問卷調查非資訊相關科系學生對於程序性程式語言與物件導向程式語言的學習差異。但是整體而言，非資訊相關科系學生的程式語言相關的實證性研究仍屬少見。

在教學方法上，過去研究對於非資訊相關科系學生學習程式設計，提出了一些改善的方法，例如採用不同的教法或不同的課程設計 (Li & Wang, 2012)，或是依照學生的程度，設計較為淺顯易懂，以區分初學與進階的教學內容 (Fatourou, Zygouris, Loukopoulos, & Stamoulis, 2018)、或是追蹤學生的學習歷程 (Shell & Soh, 2013)，以瞭解非資訊相關科系學生與資訊相關科系學生學習上的差異等建議。因此本計畫將針對非資訊相關科系學生的需求，設計一個適合的程式設計學習系統，以及專屬的教材與學習模式，並探究其學習的成效。

2. 線上與混成式課程對程式設計學習的影響

近二十年來，由於資訊科技的進步，加上開放高等教育資源（open educational resources）的理念下，線上教學在校園課程中開創了新的局面（汪耀華、張基成，2018），這二十年之間有兩個重要的線上課程的發展：包括開放課程（OpenCourseWare, OCW）與大規模網路開放課程（何榮桂，2014）。1999年麻省理工學院（MIT）提出了開放課程的構想，並於202年開放50門線上課程，使用其課程者無須註冊，MIT也不給予學習者學位與證照，也不提供其入學申請。之後為整合資源與推廣，許多提供OCW的機構成立了「開放課程聯盟」（OpenCourseWare Consortium, OCWC），成員遍及全球，台灣的交通大學也於2007年加入，並邀集國內大學成立台灣開放式課程聯盟（<http://www.tocwc.org.tw/>），目前聯盟成員學校有22所，目前線上課程超過1000門，至2018年底為止，參訪人次已將近300萬人次（<http://www.tocwc.org.tw/>），可見對國內高等教育的學術分享有其貢獻。然而OCW為完全免費提供給使用者，因此在營運上受到許多限制，課程品質也不容易管控，且內容設計上仍以傳統的文字與影音教材為主，呈現方式大多為單向無互動的教學方式，這也因此造就了磨課師的崛起（何榮桂，2014）。MOOC為加拿大愛德華王子島大學之Dave Cormier和Bryan Alexander於2008年所提出，從2012年相關的學習系統開始大量出現，例如Coursera（<https://www.coursera.org/>）、Udacity（<https://www.udacity.com/>）、edX（<https://www.edx.org/>）、Khan Academic（<https://www.khanacademy.org/>）、學堂在線、ewant育網開放教育平台等。鄭念慈（2017）整理文獻指出MOOC有以下六項特色：1. 課程打破在教室學習的時空限制、2. 學員成為學習的主人，由學員自己決定學習進度，展現主動學習的精神；3. 學習內容以小單元影片呈現，避免長時間學習降低學習成效；4. 學員可透過線上討論區隨時提出問題或補充相關資訊，解決學習者問題促進雙向溝通；5. 教材資源整合到同一平台資利於學習；6. 學生可透過網路社群相互討論學習心得和分享經驗。因此，MOOC有許多的優點，例如呂冠緯（2015）指出磨課師讓學習可以不侷限於教室，老師可以透過平台掌握學生學習進度，在課堂內無須重複講述簡單概念，而是解決學生學習問題，安排延伸與具跳躍思考之活動。基於看好磨課師的前景，我國教育部也於2013年推出為期四年（2013-2017）的數位學習推動計畫中加入了「推動磨課師」子計畫，希望透過產官學合作機制，共同發展新一代的線上開放式課程模式，提昇國內線上課程品質（何榮桂，2014）。

不過磨課師在推行之後，也慢慢出現了問題，例如2013年與Udacity合作的聖荷西州立大學（San Jose State University, SJSU）宣稱學生的學習成效不如預期，而與Udacity終止合作（劉怡甫，2014）。此外，磨課師最為人詬病的就是低完課率（Conijn et al., 2017; 胡全威，2017; 劉怡甫，2014），而且在MOOCs平台上開課，MOOCs的學生是來自不同背景又無法面對面授課，帶給教師極大的震撼與教學的挑戰（Roth, 2013）。另外，缺乏面對面的互動，時空的分隔常會造成學習者的學習孤立感，因而有大量的學生因遇到問題或學習瓶頸時，無法得到立即的協助而中輟（賴素純，莊雅茹，& 顏春煌，2014）。

因此，對磨課師有許多批評的聲浪出現，以及出現了改良的模式，例如SPOCs（黃能富，2015）與翻轉教室（黃志雄，2017; 鄭念慈，2017）等的混成式課程相繼出現，所謂混成式課程是指將網路上虛擬與真實的課程融合於教學設計中。利用這兩種學習環境的長處，強化學生整體學習的參與感與學生學習的成效（史美瑤，2014）。不過混成式學習的定義至今並無統一的說法以及界定標準（陳佩英、黃天仕、許美鈞、侯仲宸，2016），Graham（2005）於《混成學習手冊》（Handbook of Blended Learning）一書中定義混成學習為傳統面對面講授（face-to-face）與電腦輔助教學（computer-mediated instruction）的結合，以各種學習活動為基礎（event-based activities）經混合而成為一種有系統的學習方法（Singh, 2003）。顏崇凱（2015）定義「混成式學習」是將面對面教學、同步網路學習與非同步學習等兩種以上學習法混成，以突顯出學習法的優勢並足以彌補缺點，建構成一套適合學習者的學習活動；（賴素純等人（2014）認為混成式課程為老師或開課單位在課程中視教學需要，而機動選用實體教室、同步模式或非同步模式來進行教學的方式。So與Brush（2008）認為混成式學習是

指任何學習方式，包括最常見的面對面教學及非同步和／或同步的電腦科技的組合。在本研究中所介紹的混成式課程，是採用較為廣義的定義，指結合線上教學與面對面教學的課程。

由研究顯示，混成式課程有許多的優點，包括可以增進學生的參與程度（洪嘉駿、施昆易、王淑卿、董曜瑜、王國華，2018），包含學生與學生以及師生間的互動（陳芷如，2017），以及可以提高學習動機（Dantas & Kemm, 2008; Horstmann, Renninger, Hennenlotter, Horstmann, & Stenzl, 2009; Lin, Liang, & Tsai, 2012），並可以提高學習成效（Bliuc, Ellis, Goodyear, & Piggott, 2011; McCarthy, 2010）。也有許多研究指出混成式課程比傳統學習環境對學習者更有幫助（徐敏珠、楊建民，2006；劉明洲、汪冠宏，2010）；顏崇凱（2015）的研究也指出混成式學習給予學生新鮮感，相較於傳統的教室教學，學生會顯得更有活力，對於學習動機低落之學生更顯如此，而原本程度較好之學生，可以較願意自我學習，亦可依照個人的進度與需求觀看數位教材，進而提高學生學習與接觸之意願。另外，相較於單純線上的學習環境，混成的學習環境帶來的學習成效是比較高的（張基成、徐育昇，2011；Akkoyunlu & Yilmaz-Soylu, 2008）。此外，研究也發現在混成學習中，透過Facebook等線上討論空間，可以增進學習者的社會臨場感與學習滿意度（賴素純等人，2014）。另外，Owston等人（2013）的研究發現，在混成式課程中，學習動機高的學生會比學習動機弱或習慣面對面教學的學生，在學習成果上有明顯的進步。但是，也有少部份研究指出，混成學習相對於傳統實體學習，對學生的成績和批判性思考能力沒有顯著差異（Güzer & Caner, 2014）。

混成式學習的優點在於它結合了實體與網路等科技的學習，被廣泛應用在各種不同的領域（陳芷如，2017），Güzer與Caner（2014）指出學生喜歡網路的學習環境，但是並不想放棄面對面的學習機會，這顯示學習者認為網路和實體兩種學習環境都各有優點，應該同時具備。另外，史美瑤（2014）綜合文獻後指出混成學習有以下優點：1. 學生的學習成效較高；2. 增加學生與學生之間以及與老師的互動；3. 促使教師改變教學方法和教學設計；與4. 保留學生學習的過程與成果。

由上述研究結果顯示，大部份研究都對混成式教學持正面看法，不過研究也指出混成式課程會帶來許多教學上的轉變與挑戰，例如學習環境會由原本的教室，延展到教室外，以至於可以行動學習或是無所不在的學習（陳佩英等人，2016）；另外，學生學習模式的改變，使得學生的個人學習不僅從課堂轉到課前，且與教師和其他同學的互動也提早了。教師在課堂內從講課變為解決學生概念模糊之處，以及進行加深加廣之討論。對於教師而言，教師減少講課時間，然而在暑期和學期初，卻需要花費更多時間共同研討教材與備課，舉凡影片錄製，數位互動軟體的學習，課堂討論的掌握等，都必須於課前做好準備（陳佩英等人，2016）。因此開課的教師必須能夠有相當程度地掌握所使用的網路平臺，也要確定所使用的電腦科技不會造成學生學習上的阻礙。且無論教師或學生遇到問題，都確實要有支援的管道，或是教師能自己處理（史美瑤，2014）；另外，對學校而言，技術和專業發展的支持也是混成學習不可或缺的条件（Calderon, Ginsberg, & Ciabocchi, 2012），學校除了必須投入更多的經費、設備與人力之外，課程安排也會增加許多的複雜性（史美瑤，2014），以上這些問題都是在開設混成式課程時必須考慮的。

由上述研究結果得知，對學生學習而言，輔助學生能主動學習，並有良好的師生或同學間的互動，增加學習動機以提高學生主動學習的意願等因素等，是混成式教學重要的影響因素，因此本研究希望提供更多的同步、非同步，以及面對面的互動方式，在設計教材時也希望能夠針對分資訊相關科系學生的需求，多增加實用性的知識或操作，以增加其學習的動機，增加學習歷程的提醒，以維持學生持續學習活動的進行。

本研究使用的混成式教學，不但希望能提昇非資訊相關科系的學生的學習成效，也希望能夠解決學校開設大量程式設計課程的困境，因此結合線上教學與實體面對面的互動，在線上的部份，包含大部分的教學活動，例如教學影片、教學投影片與老師同步的作業講解，以及同學的作業練習與批改，與大部分的同儕討論或資源分享是在系統中完成。如此可大量減少電腦設備的運用，另外，實體的部份，學生上課或需要時可以到電腦教室練習，由教學助理

帶領學生進行面對面的實作與討論，以減少學生學習的孤獨感。由於一門課程可以有許多位助教，修課人員可以跨班級，因此修課的人數不必受一間電腦教室數量的限制，以節省師資的人員負擔。

混成式課程在各學科領域上的相關研究，除了上述列舉的之外，近幾年有許多的研究，例如洪嘉駿等人(2018)將混成式案例教學模式應用於非正式科學師資培育課程，陳芷如(2017)進行應用混成式學習於成人學習之實證經驗，呂文娟等人(2017)則進行護理新人的管路照護模組課程之混成式學習成效比較的研究，羅家駿與孫秀綺(2015)進行應用混成式學習於國小品格教育之實證研究，方之光與黃碧雲(2012)進行影像處理混成教學課程應用 Web 2.0 之行動研究，林凱胤(2015)的研究是從學習風格和科技接受模式看混成式評量策略融入教學，學科為護理科一年級學生的計算機概論，石文傑與江宗霖(2012)進行數位學習課程之製作與教學策略之應用—以「機電整合與控制」為例，蔡美惠與陳麗宇(2011)探討服務學習與數位製作混成教學法應用於國文教學之課程設計實例，Bazelais 與 Doleck(2018)進行機械學課程進行混成式課程的比較研究，Zampirolli、Goya、Pimentel 與 Kobayashi(2018)檢驗在工程教育中程式設計的混成式課程，張基成、徐郁昇(2011)則是探討高職學生電工機械混成式數位學習效果，陳育亮與鄭淑慧(2010)從事網路教學與社群學習在成人教育的應用—以混成式網路學習探討其行為意向，顧大維、張家綺與黃永欣(2009)進行數位典藏相關標準與後設資料線上混成培訓課程之設計與發展。由上述研究來看，混成式課程的應用科目相當多元，不過似乎以醫學與資訊相關課程較多，但是將混成式教學應用在非資訊相關科系學生的程式設計課程則非常少見，有待更進一步地探究。

三、研究問題

- (1)非相關科系學生在純線上與混成式之程式設計課程中，其學習動機是否有差異？
- (2)非相關科系學生在純線上與混成式之程式設計課程中，其學習滿意度是否有差異？
- (3)非相關科系學生在純線上與混成式之程式設計課程中，其程式設計自我效能是否有差異？
- (4)非相關科系學生在純線上與混成式之程式設計課程中，其學習成效是否有差異？

四、研究設計與方法

1. 研究架構與對象

本研究計畫目的比較非相關科系學生在純線上與混成式之程式設計課程中，其學習動機、學習滿意度與學習成效、以及完課率與學習行為的差異，採取不等組前後測之準實驗設計，研究設計如表2所示。修課的兩班學生以班級為單位，分配到以下兩組，混成式教學組接受混成式的教學課程，線上教學組則是純線上的課程。研究的自變項為教學模式（分成混成式課程與純線上課程等兩種），依變項包括學習成效、學習滿意度、自我效能、與學習動機。

表 2
線上與混成式課程之準實驗設計

組別	前測	實驗處理	後測
混成式教學組	O ₁	X ₁	O ₂
線上教學組	O ₃	X ₂	O ₄

X₁：表示採用「混成式的程式設計教學」。

X₂：表示採用「線上的程式設計教學」。

O₁, O₃：指實驗處理前所實施的前測，包含學習成效測驗。

O₂, O₄: 指實驗處理後實施的後測，包含學習成效測驗、學習滿意度量表、學習動機量表、與程式設計自我效能量表。

2. 程式設計學習系統

本研究所設計的程式學習系統將修改並擴充自本人科技部計畫的「同儕互動程式學習系統」而成(架構如圖1)，其中的歷程模組為新擴充的功能，互動模組需要經過修改，並增加同儕部分程式碼分享的功能，使其更適合非資訊相關科系學生進行線上與混程式的程式設計學習。

本系統以 Windows Server 2008 為作業系統，使用 ASP.net 程式語言設計實驗平台，並且使用 Microsoft SQL Server 2012 建置本系統的資料庫。本系統包含了下列四個模組，分別是出題模組、歷程模組、互動模組、及管理模組。

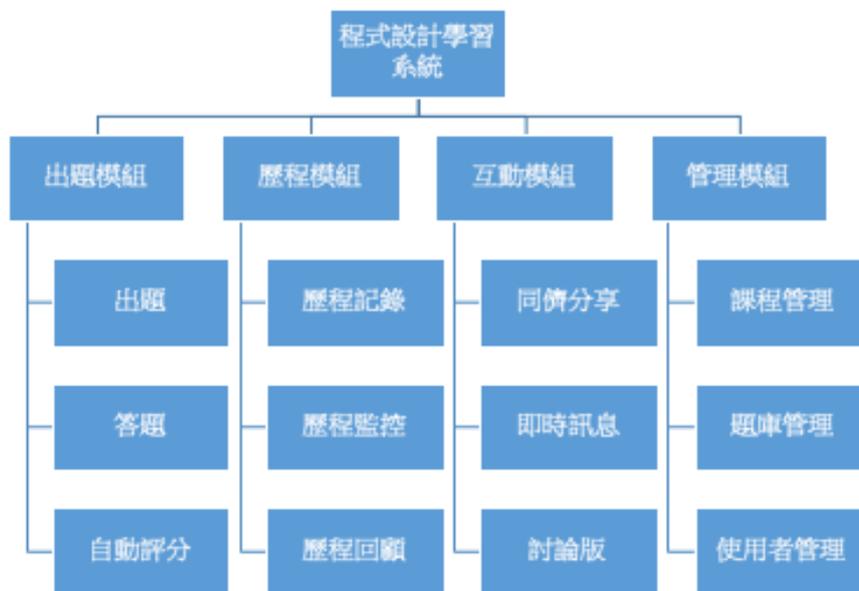


圖1、程式設計學習系統架構圖

本系統具備出題(畫面如圖2)與自動批改的功能(如圖3)，另外，也可以針對個別教材，看到全部學生的使用狀況(如圖4)。在學生端，學生在系統中可以看到自己的整個學習歷程，如圖8所示。這項功能是使用Google analytics記錄與管理學習行為的資料，再透過DataStudio軟體進行視覺化的呈現。

圖 2: 出題時題目設定的畫面

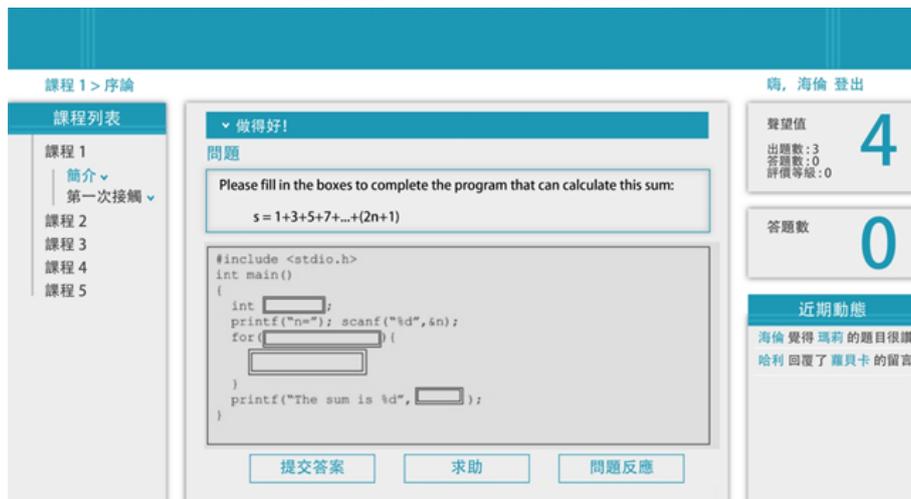


圖3、填空題的作答介面



圖4、歷程監控畫面

3. 研究工具

本研究計畫之研究工具包含以下幾個部分：

I. 程式設計自我效能量表

採用吳靜吉與程炳林 (1992) 修訂 Printrich、Smith 與 McKeachie 在 1989 年所編「自我效能」分量表，為李克特氏五點量表，量表的 Cronbach' s α 為 .926。

II. 程式設計學習動機問卷

本研究使用的「程式設計學習動機問卷」是由 Law、Lee 與 Yu (2010) 開發的動機問卷。其 Cronbach' s α 為 0.95，顯示問卷具有高度的內部相關。

III. 課程滿意度問卷：

課程滿意度為參考黃能富 (2015) 的課程滿意度編置而成，共有 9 題，為李克特氏五點量表。

五、教學暨研究成果

1. 教學過程與成果

本教學實踐研究計畫包含兩種不同的課程，分別是線上課程與混成式課程，兩種課程的教學內容皆為線上影音教材，同學可以依照自己的時間與進度，在校或在家學習，以利同學反覆觀看學習。另外，課程每週安排 2 小時討論與講解的時間（有 4 次必須參與，其餘的自由參加），進行作業講解，並讓同學有即時提問的管道。此外，期中和期末考皆集中在電腦教室，進行上機考試。其中 2 小時線上討論時間，線上課程班是在線上視訊進行，混成式課程班為到實體電腦教室中進行課程。不過由於本學期的課程，有遇到 COVID-19，所以混成式課程也有部分週次是改以線上，取代到實體教室的方式（也就是和線上課程採用一樣的方式）。

成果部分，本計畫開發了適合非資訊相關科系學生的程式設計學習教材，也根據上述教材修改部分錄製的線上教學影片，並修改完成程式設計線上學習系統，以及進行線上與混成式課程等兩班的教學實驗。資料分析後持續會將研究成果整理並投稿，目前相關成果已發表數位學習領域，包括 1 篇 TSSCI 期刊、1 篇 SCI 國際期刊，以及兩篇國際研討會論文。

2. 教師教學反思

人們對資訊科技的倚賴與日俱增，程式設計是資訊科技發展的基礎學科，更是培養運算思維的重要方法，因此愈來愈受到重視。在高等教育方面，高教深耕計畫單獨列一指標，要求大學不分學系，五年內至少半數學生在畢業前，需修過程式設計課程。本校依據這指標訂定實施辦法，要求所有新進的學士班學生需選修程式設計課程，或通過程式能力檢定，這項措施使得每年修習程式設計課程的人數暴增約 2,000 人，這對現有師資和電腦設備都是極大的考驗。因此本計畫因應這項需求，開發線上影音教材，並進行純線上與混成式的程式設計課程。

在這學期我們嘗試了不同的教學方式，將所有教材錄製成線上的影音教材，並進行純線上與混成式的程式設計課程。無獨有偶，在進行實驗課程期間，剛好有一段時間遇到 COVID-19 疫情爆發，所有課程被要求以線上的方式取代實體課程，讓學生的學習可以不受時間及空間的限制，學生可以依據自身的學習進度進行學習，這也更凸顯了本研究的重要性。不過也因此，在本實驗課程中，原先混合課程中每週 2 小時的討論與講解時間，被迫改為線上，這是本實驗稍嫌可惜之處。

3. 學生學習回饋

在本研究，學生學習的回饋包含兩個部分，第一部份是本研究針對程式設計學習，所使用的問卷，包含程式設計自我效能、學習動機、學習滿意度等量表，另一部份是學校所針對每門課程所設計的教學評量問卷，全部問卷皆為 5 點李克特氏量表，全部有效問卷為 74 份。以下針對學生在這些項目的回饋進行分析。在自我效能方面（如表一所示），以獨立樣本 t 檢定進行分析，結果顯示線上教學組 ($M = 3.06, SD = 0.86$) 與混成式教學組 ($M = 2.82, SD = 0.79$) 沒有顯著差異 ($t = 1.284$)；在學習動機方面，結果顯示線上教學組 ($M = 3.30, SD = 0.99$) 與混成式教學組 ($M = 3.30, SD = 0.86$) 沒有顯著差異 ($t = -0.008$)；在學習滿意度方面，結果顯示線上教學組 ($M = 2.98, SD = 0.92$) 與混成式教學組 ($M = 3.10, SD = 0.97$) 沒有顯著差異 ($t = -0.530$)；也就是說，兩個班級學生學生的程式設計自我效能、學習動機、學習滿意度並沒有差異。

表一、線上與混成式教學獨立樣本 t 檢定

	class	N	Mean	Std. Deviation	t value
efficacy	online	36	3.06250	.864736	1.284
	blended	38	2.81579	.787753	
motivation	online	36	3.29939	.989949	-0.008
	blended	38	3.30116	.860838	
satisfaction	online	36	2.98456	.923607	-.53
	blended	38	3.10232	.973976	

在學生學習的期末成績方面，線上教學組 ($M=72.28, SD=10.39$) 與混成式教學組 ($M=70.85, SD=13.83$) 沒有顯著差異 ($t=0.464$)。從學校的教學評量問卷結果顯示，學生在線上教學組所填答的分數平均為4.41，在混成教學組為4.37，顯示學生在這兩門課程的教學滿意度與自我學習評量的狀況大致類似。學生大部分都喜歡這兩種上課的方式，例如學生在「對於這門課我最喜歡的是」一題中的填答為「上課方式有彈性」、「採用線上方式，我比較能在狀況好的時候學習程式設計」、「超前部屬遠距教學，作業有問題可以線上提問，助教能夠快速回覆」、「老師影片製作的非常的用心，而且講解的很細，不懂還可以去問助教」等。另外，學生對課程的建議可以看出，有許多學生認為程式設計對於非相關科系的學生是有些困難的，例如：「我覺得太難了，感覺沒有很有邏輯的頭腦，需要花很多心力跟時間才能完成作業與考試，對我來說相當吃力」、「不要為難文學院的學生」、「為了通過畢業門檻我非常用心在這堂課，雖然還是很難，但發現程式也有好玩之處呀」。

六、建議與省思

2019年底爆發新型冠狀病毒肺炎 (COVID-19)，之後蔓延全世界，使得全球高教面臨嚴重考驗，絕大部分學校為了確保教學不中斷，紛紛將課堂由原先的實體教室，轉而改採線上的教學，使得線上教學成為極為重要的教學方式。此次疫情造就了視訊平台的軟硬體迅速到位，且其互動功能與普及率都有非常大的進展，師生也有較好的心理準備，以迎接這類課程。

本研究探討非資訊相關科系學生在混成式與線上程式設計課程中學習的差異，從研究結果來看，兩班學生的學習滿意度、自我效能、學習動機、與學習成效皆無顯著差異，這樣的結果也許是因為在實驗的後期，因為疫情的關係，混成式課程的實體課程部分也改為線上進行，致使兩班的學習狀況無差異。不過，整體而言，此兩種教學方式皆具有比傳統實體課程更彈性的教學方式，皆可列為未來教學可以考慮的方式，尤其是在需要大量開課，或是遇到新冠病毒疫情等特殊情況下，可以考慮部分實體、部分線上的混成課程，或是全部都採遠距的純線上課程。

針對非資訊相關科系學生的程式學習，在兩個班級中皆有部分學生認為會遇到許多困難，同樣問題也出現在實體課程中，這個問題一直是教學與研究中重要的議題，值得未來能繼續探討，期待找出一個針對非資訊相關科系學生較為理想的教學方式。

參考文獻

- Akkoyunlu, B., & Yilmaz-Soylu, M. (2008). Development of a scale on learners' views on blended learning and its implementation process. *The Internet and Higher Education*, *11*(1), 26-32.
- Bazelais, P., & Doleck, T. (2018). Blended learning and traditional learning: A comparative study of college mechanics courses. *Education and Information Technologies*, *23*(6), 2889-2900. doi:10.1007/s10639-018-9748-9
- Chang, C. K. (2014). EFFECTS OF USING ALICE AND SCRATCH IN AN INTRODUCTORY PROGRAMMING COURSE FOR CORRECTIVE INSTRUCTION. *Journal of Educational Computing Research*, *51*(2), 185-204.
- Chou, C. Y., & Sun, P. F. (2013). An educational tool for visualizing students' program tracing processes. *Computer Applications in Engineering Education*, *21*(3), 432-438.
- Conijn, R., Snijders, C., Kleingeld, A., & Matzat, U. (2017). Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS. *IEEE Transactions on Learning Technologies*, *10*(1), 17-29. doi:10.1109/TLT.2016.2616312
- Corral, J. M. R., Balcells, A. C., Estevez, A. M., Moreno, G. J., & Ramos, M. J. F. (2014). A game-based approach to the teaching of object-oriented programming languages. *Computers & Education*, *73*, 83-92.
- Dawson, J. Q., Allen, M., Campbell, A., & Valair, A. (2018). Designing an introductory programming course to improve non-majors' experiences. Paper presented at the Proceedings of the 49th *ACM Technical Symposium on Computer Science Education*, Baltimore, Maryland, USA.
- 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.
- Denny, P., Luxton-Reilly, A., Tempero, E., & Hendrickx, J. (2011). *CodeWrite: supporting student-driven practice of java*. Paper presented at the Proceedings of the 42nd *ACM Technical Symposium on Computer Science Education*. Dallas, TX, USA.
- Echeverria, L., Cobos, R., Machuca, L., & Claros, I. (2017). Using collaborative learning scenarios to teach programming to non-CS majors. *Computer Applications in Engineering Education*, *25*(5), 719-731. doi:10.1002/cae.21832
- Fatourou, E., Zygouris, N. C., Loukopoulos, T., & Stamoulis, G. I. (2018). Teaching concurrent programming concepts using Scratch in primary school: Methodology and evaluation. *International Journal of Engineering Pedagogy*, *8*(4), 89-105. doi:10.3991/ijep.v8i4.8216
- Forte, A., & Guzdial, M. (2005). Motivation and nonmajors in computer science: Identifying discrete audiences for introductory courses. *IEEE Transactions on Education*, *48*(2), 248-253. doi:10.1109/TE.2004.842924
- Garcia-Beltran, A., Tapia, S., & Sanchez-Naranjo, M. J. (2014). Performance and Professional Skills in an Online Java Programming Course for Engineering Students. *International Journal of Engineering Education*, *30*(6), 1741-1748.
- Gomes, A., & Mendes, A. J. (2007). Learning to program - difficulties and solutions. Proceedings of *International Conference on Engineering Education*.
- Gonzalez, J., Pomares, H., Damas, M., Garcia-Sanchez, P., Rodriguez-Alvarez, M., & Palomares, J. M. (2013). The Use of Video-Gaming Devices as a Motivation for Learning Embedded Systems Programming. *Ieee Transactions on Education*, *56*(2), 199-207.
- Jayaprakash, S. M., Moody, E. W. Lauria, E. J., Regan, J. R., & Baron, J. D. (2014). Early alert of academically at-risk students: An open source

- analytics initiative. *Journal of Learning Analysis*, 1(1) 6 - 47.
- Jurado, F., Redondo, M., & Ortega, M. (2014). eLearning Standards and Automatic Assessment in a Distributed Eclipse Based Environment for Learning Computer Programming. *Computer Applications in Engineering Education*, 22(4), 774-787.
- Kazakoff, E. R., & Bers, M. U. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, 50(4), 553-573.
- Lahtinen, E., Ala-Mutka, K., & Javinen, H. (2005). A study of the difficulties of novice programmers. *ACM SIGSCE Bulletin*, 37(3), 14-18.
- Li, T., & Wang, T. (2012). A unified approach to teach computational thinking for first year non-CS majors in an introductory course. *IERI Procedia*, 2, 498-503. doi:https://doi.org/10.1016/j.ieri.2012.06.123
- Liu, X. E. (2013). A new automated grading approach for computer programming. *Computer Applications in Engineering Education*, 21(3), 484-490.
- Macfadyen & Dawson, S. (2010). Mining LMS data to develop an 'early warning system' for educators: A proof of concept. *Computers and Education*, 54(2), 588 - 599.
- Martinez, L. G., Licea, G., Juarez, J. R., & Aguilar, L. (2014). Experiences Using PSP and XP to Support Teaching in Undergraduate Programming Courses. *Computer Applications in Engineering Education*, 22(3), 563-569. doi:10.1002/cae.20581
- McIver, L. (2000). The effect of Programming language on error rates of novice programmers. In A. F. Blackwell & E. Bilotta (Eds.). *Proceedings of Twelfth Annual Meeting of the Psychology of Programming Interest Group* (pp. 181-192). Corigliano Calabro. Italy: Edizioni Memoria.
- Ozan, O., & Ozarslan, Y. (2016). Video lecture watching behaviors of learners in online courses. *Educational Media International*, 53(1), 27-41. doi:10.1080/09523987.2016.1189255
- Rodrigo, M. M. T., Andallaza, T. C. S., Castro, F., Armenta, M. L. V., Dy, T. T., & Jadud, M. C. (2013). An analysis of JAVA programming behaviors, affect, perceptions, and syntax errors among low-achieving, average, and high-achieving novice programmers. *Journal of Educational Computing Research*, 49(3), 293-325.
- Roth, M. S. (2013). My modern experience teaching a MOOC. *The Chronicle of Higher Education*, 59(34), 18-21.
- Sattar, A., & Lorenzen, T. (2009). Teach Aice programming to non-majors. *SIGCSE Bull.*, 41(2), 118-121. doi:10.1145/1595453.1595488
- Shell, D. F., & Soh, L. K. (2013). Profiles of motivated self-regulation in college computer science courses: Differences in major versus required non-major courses. *Journal of Science Education and Technology*, 22(6), 899-913. doi:10.1007/s10956-013-9437-9
- Shih, Y. -F & Alessi, S. M. (1993). Mental models and transfer of learning in computer programming. *Journal of Research on Computing in Education*, 26(2), 154-175.
- Sorva, J., Karavirta, V., & Malmi, L. (2013). A Review of Generic Program Visualization Systems for Introductory Programming Education. *Acm Transactions on Computing Education*, 13(4), 64.
- Tan, J., Guo, X. P., Zheng, W. S., & Zhong, M. (2014). Case-based teaching using the Laboratory Animal System for learning C/C plus plus programming. *Computers & Education*, 77, 39-49.
- Tek, F. B., Benli, K. S., & Deveci, E. (2018). Implicit theories and self-efficacy in an introductory programming course. *IEEE Transactions on Education*, 61(3), 218-225. doi:10.1109/TE.2017.2789183
- Urquiza-Fuentes, J., & Velazquez-Iturbide, J. A. (2013). Toward the effective use

- of educational program animations: The roles of student's engagement and topic complexity. *Computers & Education*, 67, 178-192.
- Winslow, L. E. (1996). Programming pedagogy - A psychological overview. *SIGSCE Bulletin*, 28, 17-22.
- Xinogalos, S. (2016). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. *Education and Information Technologies*, 21(3), 559-588. doi:10.1007/s10639-014-9341-9
- Zampiroli, F. A., Goya, D., Pimentel, E. P., & Kobayashi, G. (2018). Evaluation process for an introductory programming course using blended learning in engineering education. *Computer Applications in Engineering Education*, 26(6), 2210-2222. doi:10.1002/cae.22029
- Zhang, J. X., Liu, L., de Pablos, P. O., & She, J. H. (2014). The Auxiliary Role of Information Technology in Teaching: Enhancing Programming Course Using Alice. *International Journal of Engineering Education*, 30(3), 560-565.
- 方之光、黃碧雲(2012)。影像處理混成教學課程應用 Web 2.0 之行動研究。[Action Research: The Design and Evaluation of Blended Learning on an Image Processing Course Using Web 2.0 Technologies]。 *教學科技與媒體*，102，41-60。
- 王令宜(2017)。美國推動電腦科學(Computer Science)教育對我國之啟示。 *教育脈動*，10。取自 <https://pulse.naer.edu.tw/Home/Content/89dea554-701f-4583-82c9-c57bc03d6341?insId=d815dc78-af80-4931-bcac-bebb539cfe9c>
- 王梅玲(2017)。大學資訊素養線上課程教學模式與策略研究 [Study on Instructional Models and Strategies of Information Literacy Online Courses for Higher Education]。 *圖資與檔案學刊*，91，20-46。doi:10.6575/jila.2017.91.02
- 史美瑤(2014)。混成學習(Blended/Hybrid Learning)的挑戰與設計。 *評鑑雙月刊*，50，34-36。
- 石文傑、& 江宗霖(2012)。數位學習課程之製作與教學策略之應用—以「機電整合與控制」為例。 *資訊科技國際期刊*，6(2)、146-154。doi:10.30111/ijait.201212.0006
- 何榮桂(2014)。大規模網路開放課程(MOOCs)的崛起與發展。[Brief History and Development of MOOCs]。 *台灣教育*，686，2-8。
- 吳靜吉、程炳林(1992)。激勵的學習策略量表之修訂。 *中國測驗學會測驗年刊*，39，59-78。
- 呂文娟、徐曼瑩、蔡衣帆、楊文理、高木榮、李俊秀(2017)。護理新人的管路照護模組課程之混成式學習成效。[The Effectiveness of Blended Learning on Tube Care among New Nurses]。 *北市醫學雜誌*，14(3)、315-323。doi:10.6200/tcmj.2017.14.3.06
- 呂冠緯(2015)。跨越翻轉教學的鴻溝。 *教育脈動*，1，65-75。
- 呂國泰(2013)。以雲端服務為基礎之即時同儕學習系統建置與應用(未出版之碩士論文)。亞洲大學，台中市。
- 汪耀華、張基成(2018)。現象學取向的磨課師教學經驗隱含之意義。[Denotations Pertaining to Teaching Experiences of Massive Open Online Courses on the Basis of the Phenomenological Perspective]。 *教育科學研究期刊*，63(1)、141-171。doi:10.6209/jories.2018.63(1).05
- 林凱胤(2015)。從學習風格和科技接受模式看混成式評量策略融入教學。[A Study on Integration of Classroom Response System into a Class from the Viewpoints of Learning Style and Technology Acceptance Model]。 *中等教育*，66(3)、138-156。doi:10.6249/se.2015.66.3.11
- 洪嘉駿、施昆易、王淑卿、董曜瑜、王國華(2018)。一個混成式案例教學模式應用於非正式科學師資培育之成效：由學習參與和學習感受觀點。[Effectiveness of Applying a Blended Case-Based Teaching Model to Informal Science Teacher Preparation: Perspectives of Learning Engagement and Learning Perception]。 *科學教育學刊*，26(2)、171-196。doi:10.6173/cjse.201806_26(2).0004
- 胡全威(2017)。教師如何克服大班授課的困境：以「演講學」課程為例。[How does a Teacher Overcome the Difficulty of Teaching Large Groups? Taking "Public Speaking" for Example]。 *嶺東通識教育研究學刊*，7(1)、57-79。
- 徐敏珠、楊建民(2006)。我國高等教育之數位學習發展策略分析。 *教育學刊*，26，191-214。
- 高教深耕計畫(2018)。 <http://iroffice.ntunhs.edu.tw/bin/home.php>

- 高等教育深耕計畫 (2018)。https://www.edu.tw/News_Content.aspx?n=D33B55D537402BA&s=333F49BA4480CC5B
- 康錦程 (2009)。引導合作學習對於國小學童學習程式設計之影響—以 KPL 為例 (未出版之碩士論文)。國立臺灣師範大學，台北市。
- 張基成、徐郁昇 (2011)。高職學生電工機械混成式數位學習效果。[The Effects of Blended Electrical-Machinery e-Learning on Vocational High School Students]。科學教育學刊，19(6)，549-579。doi:10.6173/cjse.2011.1906.04
- 郭士豪 (2011)。同儕教學法對國小學生學習 Scratch 程式設計之影響 (未出版之碩士論文)。臺北市立教育大學，臺北市。
- 陳育亮、鄭淑慧 (2010)。網路教學與社群學習在成人教育的應用—以混成式網路學習探討其行為意向。[Digital Learning and Virtual Community in Adult Education: A Study of Behavioral Intention in Blended Learning]。資訊管理學報，17(1)，177-196。doi:10.6382/jim.201001.0177
- 陳佩英、黃天仕、許美鈞、侯仲宸 (2016)。當「研究法」遇見數位學習：教與學翻轉的經驗談。數位學習科技期刊，8(1)，51-70。
- 陳芷如 (2017)。應用混成式學習於成人學習之實證經驗。[An Empirical Experience on Blended Learning Used in Adult learners]。數位與開放學習期刊，7，50-66。doi:10.6748/joel.201710_(7).03
- 陳柏翰 (2014)。探討 Facebook 使用在程式設計課程上的影響 (未出版之碩士論文)。國立虎尾科技大學，雲林縣。
- 陳婉寧 (2011)。目標導向回饋與配對程式設計對國中生 Alice 程式設計學習之影響 (未出版之碩士論文)。國立臺灣師範大學，台北市。
- 黃志雄 (2017)。翻轉教室模式在大學課程中的實踐與反思。師資培育與教師專業發展期刊，10(1)，1-32。
- 黃能富 (2015)。磨課師 (MOOCs) 與師博課 (SPOCs) 協同授課之翻轉教學法。教育脈動，1，101-110。
- 遠見雜誌 (2017.10.16)：https://www.gvm.com.tw/article.html?id=40477
- 劉正吉 (2011)。以 Scratch 同儕程式設計提升學童問題解決能力之探究 (未出版之碩士論文)。國立新竹教育大學，新竹市。
- 劉怡甫 (2014)。從 anti-MOOC 風潮談 MOOCs 轉型與 SPOCs 擅場。評鑑雙月刊(48)，36-41。
- 劉明洲、汪冠宏 (2010)。以網路虛擬實習工廠促進高職學生自我效能之研究。數位學習科技期刊，2(1)，45-59。
- 數位時代 (2018.02.13)：https://www.bnext.com.tw/article/48177/coding-learning-6-trends
- 蔡美惠、陳麗宇 (2011)。服務學習與數位製作混成教學法應用於國文教學之課程設計實例。[Application of Mixed Teaching Method of Service Learning and Digital Production to the Course Design of Chinese Teaching Practical Examples]。北商學報，20，159-174。doi:10.29871/jntcb.201107.0010
- 鄭念慈 (2017)。翻轉新視界—結合磨課師 (MOOCs) 的藝術教育。臺灣教育評論月刊，6(9)，288-293。
- 蕭俊卿 (2009)。合作學習教學法對學習成就與學習興趣影響之研究—以「程式設計」課程為例 (未出版之碩士論文)。國立彰化師範大學，彰化縣。
- 賴素純、莊雅茹、顏春煌 (2014)。Facebook 的使用對成人混成式學習之影響。[Impact of Facebook on Blended Learning for Adult Learners]。教育科學研究期刊，59(4)，65-98。doi:10.6209/jories.2014.59(4).03
- 聯合報 (2018-07-27)。https://vision.udn.com/vision/story/12379/3275582。
- 顏崇凱 (2015)。混成式學習教學實驗研究。臺灣教育評論月刊，4(7)，141-144。
- 羅家駿、孫秀綺 (2015)。應用混成式學習於國小品格教育之實證研究。[An Empirical Study of Applying Blended Learning in Character Education in an Elementary School]。Journal of Information Technology and Applications。資訊科技與應用期刊，9(1)，8-17。
- 蘇國源 (2013)。合作學習在程式設計教學上對學童學習動機與認知負荷之影 (未出版之碩士論文)。國立臺南大學，台南市。

顧大維、張家綺、黃永欣(2009)。數位典藏相關標準與後設資料線上混成培訓課程之設計與發展。教育資料與圖書館學，47(1)，97-118。