

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：PBM1100980

學門專案分類/Division：商業及管理

執行期間/Funding Period：2021.08.01 – 2022.07.31

運用多元化教學策略提升技職校院非資訊科系學生

學習程式語言動機與成效

智慧物聯網程式設計

計畫主持人(Principal Investigator)：何素美 副教授

共同主持人(Co-Principal Investigator)：無

執行機構及系所(Institution/Department/Program)：致理科技大學國際貿易系

成果報告公開日期：立即公開 延後公開

繳交報告日期(Report Submission Date)：111 年 9 月 10 日

目錄

壹、 本文.....	1
一. 研究動機與目的.....	1
二. 文獻探討.....	1
三. 研究問題.....	2
四. 研究設計與方法.....	3
五. 教學暨研究成果.....	5
(一) 教學過程與成果.....	5
(二) 教師教學反思.....	11
(三) 學生學習回饋.....	13
六. 建議與省思.....	14
貳、 參考文獻.....	16
參、 附件.....	18
附件一、學生基本資料問卷題目.....	18
附件二、學習前自我認知問卷題目.....	18
附件三、Code.org 學習成效問卷題目.....	18
附件四、Flowgorithm 學習成效問卷題目.....	19
附件五、APP Inventor 學習成效問卷題目.....	20
附件六、基本資料卡方檢定彙整表.....	21
附件七、基本資料與學習自我認知卡方檢定彙整表.....	21
附件八、基本資料與 Code.org 學習成效卡方檢定彙整表.....	22
附件九、基本資料與 Flowgorithm 學習成效卡方檢定彙整表.....	22
附件十、基本資料與 APP Inventor 學習成效卡方檢定彙整表.....	23

壹、本文

一. 研究動機與目的

技職院校的學生中普遍缺乏獨立思考、發掘問題及解決問題的能力，而非資訊背景學生對學習程式的意願低落，在許多教學評量意見回饋中發現：“對程式沒任何興趣”、“不了解學會的程式對未來有任何幫助”、“不懂學程式到底要幹嘛”、“真的常常很想認真上課，但就是聽不懂而放棄”。學生反應上課時都可以聽得懂老師教學內容，也可以依照老師教學的步驟逐一完成作業，但較難靈活運用，故在考試時，便會出現學生往往不太了解什麼時候該用哪個元件，無法融會貫通將所學的應用於考試。

在實作教學的過程中，當學生有問題時，通常不會思考為何出現錯誤訊息，也沒有興趣瞭解，大多數學生的解決方式是直接問老師，教師可能因為要解決某幾位學生的問題，而影響教師教學進度。程度低的學生跟不上學習進度，程度高的學生往往是等待老師解決所有學生的問題後，再回來上課，造成整個進度落後，這種現象對程度好的學生反而覺得樂得輕鬆，可以滑手機或上網做些別的事。在程式設計的學習過程中學生通常依照老師的步驟操作，逐一完成作業，但無法靈活運用，故在考試時，便會出現學生往往不太了解什麼時候該用哪個元件？以上種種最主要原因是學生缺乏學習動機、獨立思考、發掘問題及解決問題的能力。

因此，要積極面對這樣的困境，在授課方式和學生學習的模式上，需做相當程度的改變，課程目標和產出面都應重新檢視。如何設計多元化的教學策略，特別是非資訊背景的學生，提供系統性的授課方法，配合課程內容搭配適性的教案教法及循序漸進的教學任務，引導學生無懼程式學習，提高學習動機，讓同學將程式語言應用於商貿實務中掌握先機。

本研究目的為設計以學生學習為中心多元化的教學策略，運用 PBL 問題導向教學模式及專題分組學習，結合悅趣化學習，流程圖演練訓練學生邏輯思考能力，以提升學生學習程式設計之學習動機，讓學生不再害怕程式設計，引導學生了解許多領域都需要使用程式開發，達到啟發學生對程式與所學領域結合，進行創意發想，應用於商貿創新跨領域功能的程式需求的目的。

二. 文獻探討

當今的教育環境強調的是跨領域的學習，國際上許多國家已經將資訊程式開發列為中小學校必修的課程，對於非資訊科系的科技大學學生，第一次接觸程式語言的設計資訊課程，通常會遇到不少的學習挫折與障礙，學生對於學習的動機與未來自身所學領域的應用感覺不深刻，也不知道學了之後的用途(李亦君, 2019)，學生來自於不同的學習背景，所受的教育訓練與資訊養成也有所不同，對於學習的誘因就需要仰賴不同的課程變化設計，用來提升學生學習動機。Wager(2003)認為學習成效是指教師教學所期望得到的成效，並可發展成為成績評比的依據，也可反映教學設計的優劣，讓學生了解老師對此次教學活動的期待。溫廷宇 (2011)認為，學習成效是指學習者的學習成績，包括形成性與總結性評量的結果。

在(何昱穎, 張智凱, & 劉寶鈞, 2010)一文中提到興趣、相關性、期望和結果都會影響學生的學習成效(Reigeluth, 2013)，其中學生的焦慮是造成學習反效果(Young, 1991)主要因素。反

之，若能克服焦慮及讓學生產生學習的內在學習動機(Ryan & Deci, 2000)，將都能提高非電腦科系的學生，學習電腦語言撰寫學習的成效。為了降低學生的焦慮及提高內在的學習動機，本文作以下幾個學習理論相關文獻探討，用來加強本研究的理論根據：

以學生為中心的翻轉教學：傳統的教學教室都是以老師為主的授課，對於台下的學生就是以接收課程的內容為主，老師往往不知道學生是否有真正了解，或對上課內容理解程度或興趣狀況。透過以學生為主的討論及合作，解決老師所出的問題或難理解的部分(黃政傑, 2014)，同學間透過討論合作，課前準備及隨時上網找資料並討論主題等，化解學生對問題的焦慮感，進而提高學習成效與學習動機。拜科技之賜，手機的發達促成了翻轉教學的興盛(Berrett, 2012; Frydenberg, 2013)，透過翻轉教學讓老師更了解學生的能力與學習所呈現的互動及時效果，也讓學生達到朝著想學的方向與對問題的瞭解更加透徹，透過此反轉學習的方向找到更適合的解決方式(黃政傑, 2014)。當今的多媒體科技發展提升了目前的數位化學習多元化，一開始程式設計課程多個研究導入悅趣化學習讓學生對課程產生好奇、愉悅，甚至融入劇情，用來提升學生學習動機及學習成效(Kiili, 2005; Prensky, 2003)。

透過問題導向學習培養運算思維能力：透過電腦的運算思維模式(Wing, 2006)讓學生學習到電腦系統如何設計及運行，幫助人類提升工作效率，同時學生可以理解電腦解決問題的方式。電腦思維能力也是現在學生更需要具備的關鍵技術(Barr, Harrison, & Conery, 2011)。一般在短時間內，對於非資訊科系學生，其實很難理解電腦的概念，但可以透過學習寫程式協助非專業的學生理解電腦邏輯運算架構(Cooper & Dann, 2015; Grover & Pea, 2013)，同學在課堂上經由老師提出的問題來學習，透過群體討論方式，上課老師鼓勵學生思考與討論並用創意來解決問題，藉由寫程式的邏輯運算思維解決老師上課出的問題，甚至應用到現實生活中解決困難的問題(Hmelo, Gotterer, & Bransford, 1997)，呼應了結合科學、技術、工程、及數學等的 STEM (Science, Technology, Engineering, and Mathematics) 整合學習(林育慈 & 吳正己, 2016)。結合兩者的教學應用，學生可以學到更踏實的運算思維解決問題的方法。

評量反饋驗證學習成效並做階段性調整：授課老師一旦採用翻轉教學以學生為學習中心的方式，來解決老師所提的問題導向學習解決問題，階段性的學習成效定時檢驗與反饋是有其必要的。學生一開始學習就先透過注意(Attention)、關聯(Relevance)、信心(Confidence)和滿足(Satisfaction)的 ARCS 檢測量表設計最佳的課程(Keller, 1987)，透過學生基本特質與邏輯思維學習成效做比對說明其學習成效正相關(蔡政宏, 2019)。在學習過程中可以透過定性定量分析及內容分析法，來了解特定的學習過程與成效以及邏輯思維的應用(Mayring, 2000; Salinger, Plonka, & Prechelt, 2008)。藉由 Rubric 評分計分表的方式，讓學生有一致性的評分標準，也方便上課老師做學生的學習狀況了解並提升課程滿意度(溫子欣, 2018)。

三. 研究問題

個人觀察所任教的技職院校的學生中普遍缺乏獨立思考、發掘問題及解決問題的能力，在程式設計的教學經驗中，發現教授私立科技大學非資訊背景學生程式力是極具挑戰性。大部份學生對學習程式的意願低落，有些學生甚至覺得對程式沒任何興趣，也不覺得學會的程式對未來有任何幫助、不懂學程式到底要幹嘛、大部份學生以前並沒有學過，真的常常很想認真上課，但就是聽不懂而放棄。在學習的過程中學生反應上課時都可以聽得懂老師教學內

容，也都依照老師教學的步驟逐一完成作業，但較難靈活運用，故在考試時，便會出現學生不太了解什麼時候該用哪個元件，無法融會貫通將所學的應用於考試。另一方面，對面少子化現況，招收的學生程度將會越來越差，對於低學習成就及學習意願較低學生，如何提升學生學習動機，乃是教師們都需努力找尋的方法。

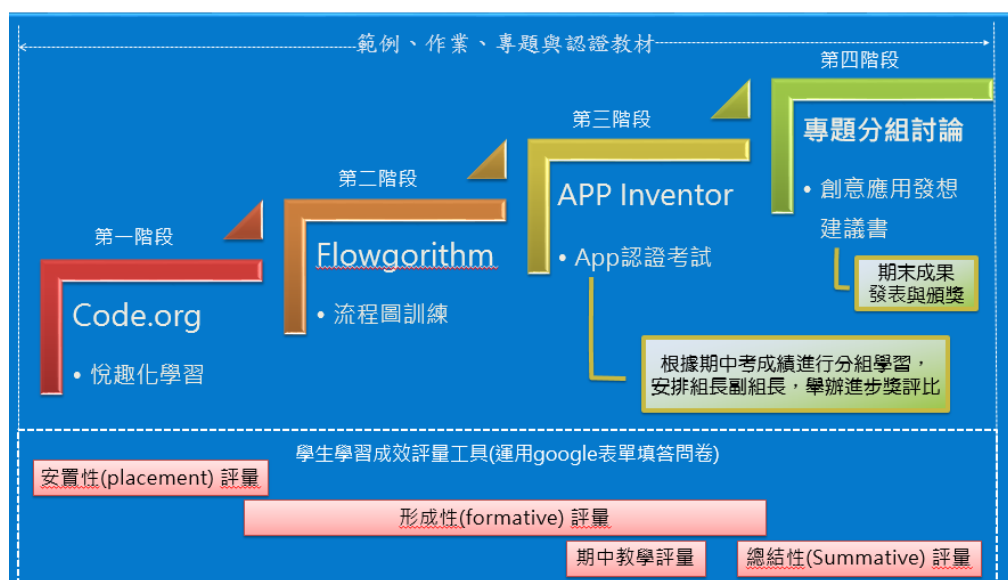
因此，設計多元化的教學策略，特別是非資訊背景的學生，提供系統性的授課方法，配合課程內容搭配適性的教案教法及循序漸進的教學任務，引導學生無懼程式學習，提高學習動機及興趣，進一步提升學生學習成效。本計畫研究問題如下：

1. 進行課程規劃與教案設計，運用範例、作業、專題與認證教材透過分組學習共同完成遊戲任務及流程圖繪製，可培養同學無懼跨領域學習的積極態度。
2. 課堂練習程式設計各項元件運用，再搭配階段性的課堂作業演練，可習得程式設計架構及概念。
3. 透過取得 APP 證照及完成「創意應用發想建議書」，強化學習成效。
4. 各教學階段完成後，輔以學習成效評量，以探討學生學習反饋意見，做為教學改進之依據。

四. 研究設計與方法

1. 教學設計

本研究對象為本校國際貿易系（以下簡稱國貿系）及應日系（以下簡稱應日系）四技部一年級修讀「智慧物聯網程式設計」之學生共 2 班修課人數 116 人。設計以學生學習為中心多元化的教學策略，運用 PBL 問題導向教學模式及專題分組學習，結合悅趣化學習，流程圖演練訓練學生邏輯思考能力，以提升學生學習程式設計之學習動機，運用範例、作業、專題與認證教材透過分組學習共同完成創意應用發想建議書，課程中錄製所有教學過程影片，放置於本校致理數位學院中，讓學生可於課後進行復習，並透過以下四個階段的執行，並於各階段執行完畢後進行各階段成效評量最後進行國貿系與應日系學生學習態度與認證考試成效差異分析，如下圖所示：



第一階段：教學開始實施安置性評量(學習前自我認知問卷)以了解學生的知識、技能與學習特質，透過 Code.org (<https://code.org/>) 提供的多種遊戲軟體，結合卡通及動畫，只需利用程式邏輯，把「程式方塊」依照需求決定擺放順序，組成不同的解決方案來解決不同的難題，最後完成遊戲，活潑的內容讓學生能從遊戲中學程式，進而讓學生對開發軟體 APP 不懼怕產生興趣。

第二階段：運用問題導向教學方法，改變傳統學習程式之教學方法，先透過 Flowgorithm (<http://www.flowgorithm.org/>)免費流程圖軟體工具訓練學生結構化思考及邏輯分析的能力，可以獨自完成理解問題並完成對應的流程圖，執行形成性評量作為判斷學生及老師在教學上的不足，以補充改正以利學生在學習上及時改善。

第三階段：教授 APP Inventor 基礎概念、邏輯運算、迴圈、及判斷式並訓練其可以獨立完成對應的程式，透過每次上課作業繳交情形與作業成績，適時了解學生學習上困難的部分，利用課後輔導補教的機會來協助學生克服學習上的困難。

第四階段：專題分組討論創意應用發想建議書及輔導學生 App 認證考試以驗證學習成效，進行總結性評量評估整個教學成效的總結確認是否達到所設定的目標。

本課程學生學習成效評量，分為安置性評量（10%）、形成性(formative)評量（40%）及總結性(summative)評量（50%），安置性評量於上課第一週進行測驗，形成性(formative)評量包含：各階段成效評量、平時作業、學生出席率、期中教學反應回饋問卷等。總結性(summative)評量之量化為期中上機測驗、第 17 週證照考試及期末教學反應回饋問卷，質化評量為第 18 週「創意應用發想建議書」及期末學習心得回饋。

2. 各階段問卷設計操作型定義

- (1) 安置性(placement)評量：包含了解學生基本資料及學習前自我認知等。
 - A. 學生基本資料：性別、高中就讀科系、入學管道、學習特質及電腦學習經歷，內容請參考附件一。
 - B. 學習前自我認知：分別為學習態度、程式力期待與認知、資訊科技接受度、自我期許與認知以及學習本課程期待等共 14 題，內容請參考附件二。
- (2) 形成性(formative)評量：除了平時作業成績、學生出席率、期中教學反應回饋問卷，各階段成效評量包含下述三種軟體學習成效評量。
 - A. Code.org 分為學習動機與興趣、使用介面及環境、學習態度以及學習成效四個構面等共 25 題，內容請參考附件三。
 - B. Flowgorithm 分為學習動機與興趣、使用介面及環境、學習態度以及學習成效四個構面等共 24 題，內容請參考附件四。
 - C. APP Inventor 分為分別為學習動機與興趣、使用介面及環境、學習態度、學習成效以及教學內容與方法等共 25 題，內容請參考附件五。

(3) 總結性(Summative)評量：

期中及期末教學反應回饋問卷量表為申請人任職學校之教學反應回饋問卷，教學反應回饋問卷的題項共計 26 題，問卷的內容可依據性質分類為「學生自我學習狀況」、「整

體評量」、「教學內涵」、「教學態度」、「教師特質」、「學習成效」與「程式力的提升」七個層面。

3. 衡量方法

問卷項目參考 Fournier(2001)、Jong-Won Park, Kyeong-Heui Kim, JungKeun Kim,(2002)、陳玉屏(2005)修訂而成，採李克特五點量表(Likert Scale)，以李克特五點量表(Likert Scale)：「非常不同意」、「不同意」、「普通」、「同意」與「非常同意」五等量表填寫，正向陳述之答案越正向分數越高（如：「非常同意」為 5 分，「同意」為 4 分，以此類推）；反之，負向陳述答案越負向分數越高（如：「非常不同意」為 5 分，「不同意」為 4 分，以此類推）。

4. 資料處理與分析

(1) 量性分析：以統計軟體 SPSS 23.0 for Windows 進行建檔，並進行以下分析：

- A. 所有研究對象之基本屬性與學習成效評量資料進行信度分析，以描述性統計，包括：個數、百分比、平均值、標準差、範圍等進行分析。
- B. 整理各階段評量問卷內容，分析各階段卡方檢定分析。
- C. 針對安置性評量問卷結果（前測）與總結性評量之結果（後測）進行分析，探討了解學生學習動機及學習成效提升情形。
- D. 課程之期中及期末教學反應回饋問卷及結果，進行分析比較：國貿系學生及應日系學生在學習上的異同。
- E. 針對 109(實驗前)與 110 學年度(實驗後)之學習成效進行比較

(2) 質性分析：以學生的平時作業、教師教學日誌、學生訪談紀錄、階段評量問卷質性問題及期末學習反思心得回饋為分析對象，進行分析，以瞭解程式設計之學習進度、解決問題能力、自主學習、面對跨領域問題的態度，探討此教學模式學生能力成長及學習感受之差異等。

五. 教學暨研究成果

(一) 教學過程與成果

本研究實施程序分為開課前、開課中與開課後。

開課前：完成教材中的教案設計、各階段學習成效評量量表。

開課中：第一週說明課程規劃、安排分組合作學習、進行課前安置性評量；隨時由學生課堂反饋檢討課程實踐方式；期中考後進行重新分組合作學習，根據期中考成績進行常態分配，將每班學生分為 7 組，每組 7~8 人，各組期中考成績總平均分數落差不大於 1，指定最高分為組長及次高分為副組長協助同學解決問題與困難，最後三週進行教學反應回饋問卷、證照考試、辦理「創意應用發想建議書」觀摩分享及頒獎，為鼓勵及感謝組長副組長的協助頒發組長副組長協助同組學生獎品，分組進步前二名獎品以提升學生學習動機，建議書競賽前三名獎品。

開課後：平時上機實作時，教師時常到同學的座位中實際了解學生實作情形，即時解決有問題同學的問題，並針對學習落後學生進行個別輔導，以協助學生學習。同時安排組長及副組長於練習中主動關心並協助有困難的學生完成作業，大部份學生均能於下課前將作業做完並繳交。各階段學習成效評量量表共有 114 位同學填答問卷，其中扣除填答不完整之無效問卷 6 份，有效問卷為 108 份。

1. 期中考試

期中考利用 Open Book 進行，考題設計上只要求學生會應用故與平時作業的題目類似，將平時練習綜合應用於不同領域中。例如：運用 if 條件式判斷男生女生後，再其計算標準體重；單迴圈則教授累加概念，考試則考累乘或減法可減次數；雙迴圈則教九九乘法表，考試則將九九乘法表的顯示結果改為圖形如星號(*)取代或改成繪製不同的圖形如菱形。考試與作業最大的差異點是平時作業老師操作一段落後，再讓學生做，逐步帶領，學生可將老師授課內容用手機錄影下來，學生有問題可看影片、請教同學或老師，而期中考僅給題目，沒有任何提示如何做，不能問同學，但可參考之前作業內容或錄影檔。期中考題目設計畫面設計占 40 分及程式碼設計占 60 分，最後成績分佈結果為，40 分(不含)以下有 2%，超過 30% 學生期中成績不及格，80 分以上的有 19%(含 90 分以上有 4%)，表示大部份的同學在畫面設計上已能完全掌握，程式碼的學習仍有待加強。

2. 學習成效評量問卷分析

- (1) 基本資料分析：修課學生女同學占 71.3%，男同學 28.7%；國貿系學生為 52 人(48.1%)；應日系學生為 56 人(51.9%)；透過甄試入學的學生最多 50 人(46.3%)、聯合登記分發 29 人(26.9%)次之、技優甄選學生 22 人(20.4%)、高中申請入學 5 人(4.6%)、繁星計劃 2 人(1.9%)；高中職就讀應用外語科為最多 35 人 (32.4%)、資料處理科 32 人 (29.6%)次之、商業經營科 13 人(12%)、國際貿易科和餐旅觀光科皆為 10 人(9.3%)、綜合高中 3 人(2.8%)、一般高中 2 人(1.9%)、會計科、園藝科和電子商務科皆為 1 人(0.9%)；99.1% 學生有自己的電腦。
- (2) 學習前自我認知：有 73.2% 的受訪者上課都覺得自己不會看其他科目的書或玩手機，有近 50% 受訪者遇到較困難的學習內容時，會放棄或只學習比較簡單的部分。有 47% 曾使用過或學過程式設計，曾使用過的程式設計工具包括 Scratch, App Inventor, Visual Basic, Python, Java, JavaScript, C, C++ 等，95% 的同學有屬於自己的電腦，87% 同學覺得資訊科技對生活有很大的影響，82% 同學覺得在未來有很多目前的工作被機器人或人工智慧取代，65% 同學認為將程式設計相關課程的內容學好是重要的，27% 同學在學習程式設計時會感到焦慮，只有 17% 同學對寫程式很有興趣。
- (3) 不同系所學生特質分析
 - A. 國貿系集中在商科如資料處理科、國貿科、商經科及應用外語科，只有一位電子商務科；國貿系以甄試入學 46.20% 最多，其次是技優甄審 28.80%；高中以前曾學過程式占 57.69%；使用電腦及手機時間以 0~3 小時最多占半數以上；使用手機玩遊戲時間以 0~3 小時最多占六成以上；使用手機上網時間以 4~6 小時最多占五成；玩手遊 0~3 小時最多占六成以上，超過 7 小時以上 17.2%。

B. 應日系學生較為多元，集中應用外語科、餐旅觀光科、商經科及資料處理科，少數有綜合高中、園藝科、會計科及一般高中；甄試入學 46.40% 最多，登記分發 30.40%，高中申請入學有 8.90%；高中以前曾學過程式僅有 39.29%；使用電腦及手機時間以 0~3 小時 41.10% 及 4~6 小時 39.30% 較多；手機上網集中在 4~9 小時占近七成；玩手遊 0~3 小時為 54.60%，超過 7 小時以上學生占 21.5%；

(4) 三種軟體課程同意程度平均數分析：整體而言，Code.org 滿意程度最高，平均數達 4.16，因 Code.org 提供的多種遊戲軟體，結合卡通及動畫，只需利用程式邏輯，把「程式方塊」依照需求決定擺放順序，組成不同的解決方案來解決不同難題，最後完成遊戲。APP Inventor 次之，平均數為 3.94，接近滿意，在透過不斷的邏輯運算、迴圈、及判斷式的訓練，讓學生可以獨力完成對應的程式，已經操作過 Code.org 對 APP Inventor 操作不再感到害怕，覺得不是那麼的困難而有信心。最後是 Flowgorithm 課程，平均數為 3.77，利用 Flowgorithm 流程圖軟體工具來訓練學生結構化思考及邏輯分析的能力時，學生需要更多的時間來理解邏輯概念，才能獨自完成理解問題並完成對應的流程圖，因此，導致學生滿意度較差。

3. 信度分析

本研究共學習前自我認知、Code.org 學習成效評量、Flowgorithm 學習成效評量及 App Inventor 學習成效評量等四份問卷，進行整體信度分析，Cronbach's Alpha 值為 0.952，另分別針對四份問卷進行信度分析，其 Cronbach's Alpha 值分別為 0.723、0.910、0.953 及 0.946。問卷回收僅有 108 份，由於樣本數較少，為能有效進行卡方檢定，故進行資料合併，將「非常同意」及「同意」合併為「同意」，為 3 分，「普通」為 2 分，「非常不同意」及「不同意」合併「不同意」為 1 分。此外亦將部份題項選項合併以利資料分析，說明如下：

- (1) 高中職就讀科系歸納為三類：商管群（商經科、國貿科、會計科及電子商務科）、資管群（資料處理科、資訊科及資處科）及語文與其他（外語類、園藝科、餐旅群、餐飲科、觀光科一般高中、綜合高中及香港的普通中學）
- (2) 入學管道歸納為三類：將技優甄審、高中申請入學及繁星計畫合併為一類
- (3) 使用電腦或手機時間 7~9 小時、7~9 小時及 13 小時以上合併為 7 小時以上

4. 卡方檢定

- (1) 基本資料各題項卡方檢定：7 題達顯著水準，18 項卡方檢定有顯著差異，如附件六。
- (2) 基本資料與學習自我認知：8 題達顯著差異，17 項卡方檢定有顯著差異，如附件七。
- (3) 基本資料與 Code.org 卡方檢定：8 題達顯著水準，20 項有顯著差異，如附件八。
- (4) 基本資料與 Flowgorithm 卡方檢定：7 題達顯著水準，20 項有顯著差異，如附件九。
- (5) 基本資料與 APP Inventor 卡方檢定：9 題達顯著水準，47 項有顯著差異，如附件十。

5. 「創意應用發想建議書」成果觀摩與分享

證照考試後到期末考週之間進行創意發想建議書的討論及撰寫，並邀請業師進行評分，兩個班級共產出 27 份創意發想建議書，下表為業師評量成績及評語，並於期末考當週舉辦分享會，讓所有的同學彼此觀摩各組的創意，最後頒發前三名獎狀及獎金。

組別	主題	成績	名次	評語
1.1	JT tag	86.94	2	1.主題想解決的問題明確，可再加強具體化呈現 2.從旅遊書電子化發想，與書商合作，結合電子地圖與流行資訊，企劃完整且具可行性
1.2	1 不會日語也能在日本趴趴走	80.8	15	1.結合一些現有及創新技術之整合性方案，實用性不錯 2.以手機實現"AR 擴增實境 AI 功能"不會比傳統手機導航省電，智慧眼鏡目前也尚未普及化，提案展現出對未來生活情境的合理想像，然缺乏技術掌握度的自信無助於提升方案可行性
1.3	團購批發小舖 APP	72	24	1.企劃架構完整，流程規劃明確，較像可實行之方案 2.實用且可行的提案。如以團主為主要使用者，或可考慮增加"廠商管理"相關功能
1.4	幫你一把 (雨傘租借 app)	85.34	3	1.環保立意佳，可再加強實務上傘如何管理之規劃 2.租借雨傘的機制亦可適用於其他具共享性質的物品。然租借雨傘的經濟效益宜進一步試算，瞭解為了環保需要付出多大的固定成本
2.1	Uber eats 的改善建議書	88.54	1	1.針對市場現有方案之缺失進行分析，並具體提出改善方案建議，企劃完整性佳 2.雖有頗多錯別字，但還能理解提案內容
2.2	日語試試看	84.54	5	1.以自身經驗發想，尚稱不錯 2.以遊戲的方式學習，難度要適中，才有持續的成效
2.3	串流音樂平台 (音樂播放器)	81.34	12	1.市面上已有許多類似方案，宜強化企劃產品之特色 2.有趣但不易實作的提案，各大熱門音樂播放器不一定樂見來蹭熱度的"串流音樂入口 App"
2.4	行動電源	62	27	行動電源租借站有應用層面之想法，可惜無實際具體落實之策略方案
3.1	不求大神給番號自己來當老司機	72	24	1.文詞略顯浮誇，實際作法規劃可再具體些，錯別字宜多校稿 2.Google 以圖搜圖還有許多可改進空間，但"GOOGLE...以圖搜圖的功能被移除"?請試試 https://www.google.com.tw/imghp 或於 Google 搜尋頁面點選右上角 Gmail 旁的"圖片"
3.2	英語學習程式	76.54	22	1.針對市面現有產品，新增較優之功能，實用性可 2.從取代教科書發想會實用；"將選詞改成拼字的方式進行測驗"從選擇字詞改為更高難度且不易在手機上操作的拼字輸入，若無助於提升 App 使用率，增加難度並不等於有效的創意
3.3	記帳理財	83.6	7	1.企劃架構清楚具體 2.銀行因資安考量並不樂於開放帳務資訊給第三方存取。記帳理財目標宜從利用手機簡化登錄收支明細著手，勿執著於連結銀行帳戶
3.4	行動點餐結帳機	73.86	23	1. 概念可行，雖市面上已有類似方案，但企劃中對流程及操作均有考量到，可再強化特色

				2.提案可行，但相較於已逐漸普遍的自助式點餐系統（例如隨桌電子點餐裝置以及速食店的自助點餐機）創意不足
4.1	聲控支付系統	79.86	18	1.創意發想不錯，宜加強技術可行性之分析與規劃 2.若基於防疫考量，是不是更應該少開口？
4.2	租借電動機車	72	24	1.針對標的公司之服務，描述清楚、架構清晰、分析深入且量化，企劃完整度高，堪稱具體而微之小論文 2.圖表的編號不連續，檔案中找不到圖 1。企劃架構完整，但排版方式與錯別字有很大改善空間
4.3	現在吃什麼	85.06	4	1.想解決的主題明確，宜加強實際作法之規劃 2.實用但因涉及個人口味，恐難獲好評。若加入個人用餐經驗回饋，或可推薦出客製化的用餐地點
5.1	手機行銷網路比價	79.86	18	1.宜加強與現有比價網站之差異，以凸顯改良處之優勢 2."蒐集各牌子比價網站的商品去做比較"能呈現更精簡的資訊，但也可能濾除使用者感興趣的資訊。從協助使用者選擇最合適的手機發想，會比追求最低價更實用
5.2	日文慣用語字典 APP	79.6	20	1.想滿足之使用需求明確，但宜加強企劃之內容與如何實現方式 2.宜呈現與 Google 搜尋結果的差異
5.3	悠遊卡與一卡通結合	80.8	17	1.實用性佳，但宜深入分析兩者結合所所需解決之商業及技術面等問題，並試著提出解決方案建議 2.商業利益無共識是悠遊卡與一卡通優惠不一致的主因。依地理位置自動選擇可獲得最大優惠的（電子）卡片會比較可能實現
5.4	FoodPanda 升級!取消訂單不用怕!	80.94	14	1.雖主要目標（如何避免訂單被取消）不一定容易實現，但企劃次要目標（被棄單之食物轉送社福機構）倒是立意良善，具可行性 2.將具保鮮期限的餐點轉送社福對象，對外送平台能帶來什麼效益？臨時取消訂單的情形並不會因而減少。取消訂單若未出餐，並不需要轉送餐點；若已出餐，目前外送平台由外送員「自行處理」的方式更具經濟效益
6.1	Google Maps	82.54	10	1.針對現有產品之改善方向清楚，企劃架構尚稱完整，惟錯別字宜多校稿 2.「不浪費太多時間到達想去的地方」是很好的發想目標，然而 Google Map 的導航功能並不只是單純的軟體功能，還須依賴地圖資訊(須定期更新)以及即時回傳的路況資訊才能實現
6.2	日本文化懶人包	80.8	15	1.目標及企劃內容雖簡單但明確，具技術可行性 2.文化具多元呈現的特性，僅從數位化面相難窺全貌
6.3	日本各地名產總匯	84	6	1.目標及企劃內容雖簡單但明確，具技術可行性 2.從地理位置建立對應名產的關聯性是各地方政府觀光宣導單位會提供的基本資訊，且名產不一定只在原產地可購得，提案有趣但不一定實用

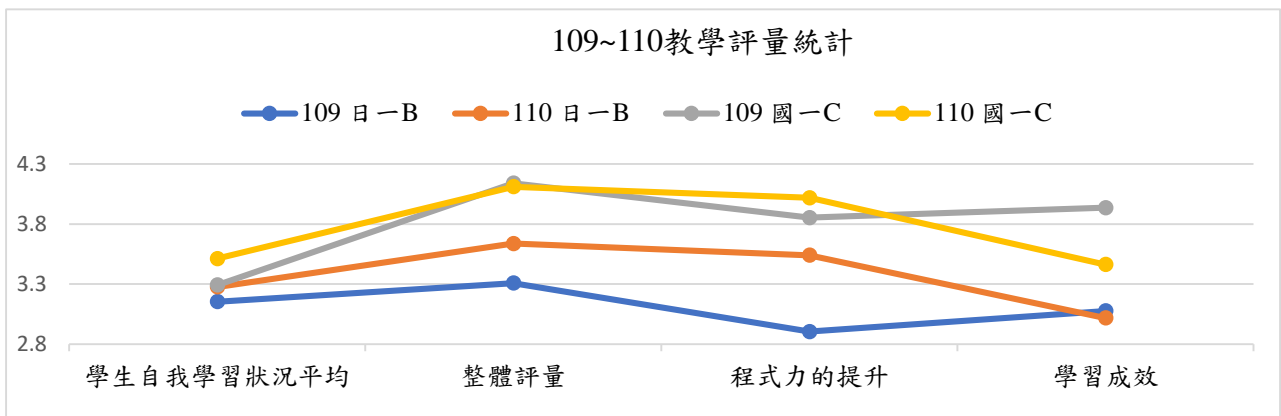
6.4	診所預約掛號系統	83.06	8	1.實用性佳，企劃架構尚可 2.各級醫療單位需求的掛號系統複雜度不同，小診所資訊化程度與大醫院也有顯著差距，從疫苗預約系統即可體會本案實作上的難度
7.1	50音學習工具	81.86	11	1.目標明確，但企劃內容及作法宜更具體化 2.實用且可行的提案
7.2	早餐，找餐	82.8	9	1.企劃架構具體，方案具可行性 2."找到我的餐點並給老闆錢"的應用情境意指只處理資訊流，不考慮金流數位化?這相對於使用外送服務但自取餐的既有 App 會有顯著的功能差異
7.3	查詢日本打工留學資訊 app	77.6	21	1.目標明確，但企劃內容及作法宜更具體化 2.惜未進一步說明如何實現"遊學和打工機會結合"的關鍵要素，此概念亦可適用於尋找合適的一般兼職工作
7.4	深入了解星巴克	81.34	12	1.主要在描述現有廠商狀況，但少了企劃應有之規劃與解決方案 2.宜進一步說明"大約 50 萬新台幣"預估費用要用於那些改善項目

6. 頒獎：舉辦「創意應用發想建議書」競賽，邀請業師評分，頒發建議書競賽前三名獎品及進步前三名獎品

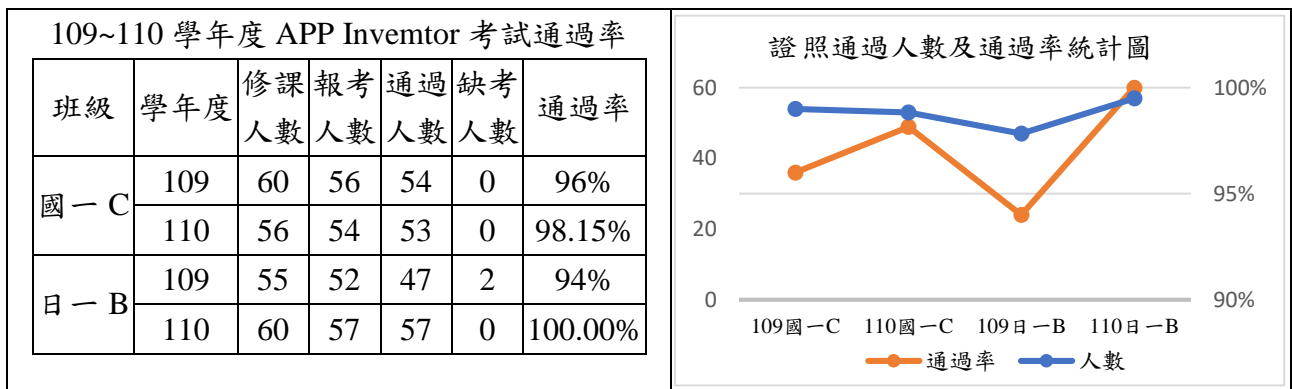


7. 期末教學反應回饋問卷及結果

109 學年度與 110 學年度教師教學評量統計，在學生自我學習狀況、程式力提升、學習成效及整體評量均有提升。



8. APP Inventor 考試通過率



9. 教學成果

(1) 已完成之教學成果

- A. 發展技職學生非資訊背景學生程式設計課程教學教材及教案 1 案。
- B. 完成 10 份課程教學影音教材並放置於 youtube 上，供修課學生隨時復習。
- C. 輔導學生考取 APP 證照，證照通過率達 99%。
- D. 各階段學習成效評量量表。

(2) 預期達成之與學生學習相關之目標

- A. 學生接受此創新教學模式後所獲得之程式設計應用能力、自主學習能力與邏輯思考能力優於僅接受傳統教學法學生，較不害怕學習程式設計，具備程式設計基礎能力，進行課前課後程式力問卷顯著進步。
- B. 培養學生獨立思考及問題解決能力，完成 Flowgorithm 流程圖繪製。
- C. 取得 App 證照，共取得 110 張證照。
- D. 結合程式應用所學領域之創新應用能力，完成 27 份「創意應用發想建議書」。

(二) 教師教學反思

本教學實踐研究計畫在課程安排了一系列教學活動，各單元時程非常緊湊，採用 PBL 問題導向式的教學方式，以實務問題為核心，鼓勵學生進行小組討論，以培養學生主動學習、批判思考和問題解決能力，除了教授 APP Inventor 並輔導學生取得 App 證照外，安排 Code.org 課程藉以引起學生學習程式動機；導入 Flowgorithm 軟體學習，以訓練學生邏輯思

維、強化程式觀念的理解及瞭解電腦的運作原理，最後透過分組討論完成「創意應用發想建議書」以培養跨領域的思考能力。但在這個最後階段學生可以參與創意應用發想討論的時間不多，在時間有限的情形下，為了避免部份學生搭便車之狀況將原先的 7~8 位一組再拆成 2 小組，每組 3~4 位，但仍然無法避免有同學搭便車的情形，同學為了能順利完成教學任務，而利用其他課程的報告稍作改寫，而失去了參與腦力激盪及創意應用發想的機會。其他教學反思如下：

1. 教學設計面：調查發現 30.5% 的學生對寫程式沒有興趣，只有 16.6% 的學生有興趣，因此在程式設計的推動仍需有努力空間，在修習完成本課程後進行課程滿意度調查發現非資訊科技系所之學生，能夠有近半數的同學不排斥學習程式設計，表示此課程的安排學生是可以接受的，有 49.1% 學生希望以後有機會可學習更多的程式設計課程。三個軟體的學習上，說明如下：
 - (1) Code.org 提供的多種遊戲軟體，結合卡通及動畫，只需利用程式邏輯，把「程式方塊」依照需求決定擺放順序，組成不同的解決方案來解決不同的難題，最後完成遊戲，活潑的內容讓學生能從遊戲中學程式，進而讓學生對開發軟體 APP 不懼怕產生興趣，故滿意 Code.org 課程的學習平均數達 4.16，有 78.7% 的同學滿意此課程的學習，同時有 72.2% 的學生覺得完成 Code.org 程式很有成就感，有六成的同學 (60.2%) 希望以後有機會可學習更多的程式設計課程，不論有沒有學過程式設計至少有六成的同學同意 Code.org 很有趣，有學過同意比例 80.4% 高於沒有學過 61.4%，但沒有學過不同意比例為 0，有學過則有 3.9% 不同意，可能原因是因為 Code.org 也適合給國中小的學生使用，因此部份有學過的大學生會覺得太簡單感到無聊。因此，可針對有學過的同學給予難度較高的任務，而非完成基本要求就好。
 - (2) Flowgorithm 程式設計課程的學習滿意度最低，平均數僅有 3.77，僅有 56.4% 學生感到滿意。69.5% 的同學同意學習 Flowgorithm 後覺得比較理解電腦的運作，可見在課程中安排 Flowgorithm 流程圖課程對有助於理解電腦的運作。46.3% 學生在學習 Flowgorithm 會感到緊張，學生覺得此課程有些難度，因此在學習上感到壓力而有緊張的感受，很多學生之前不曾受過邏輯的訓練，有可能利用 Flowgorithm 流程圖軟體工具來訓練學生結構化思考及邏輯分析的能力時，學生需要更多的時間來理解邏輯概念，才能獨自完成理解問題並完成對應的流程圖。因此在流程圖練習的需要更多時間，故可在時程安排上可再延長，讓學生多熟悉，
 - (3) APP Inventor 程式設計課程滿意度次之，平均數為 3.94，接近滿意，在透過不斷的邏輯運算、迴圈、及判斷式的訓練，讓學生可以獨力完成對應的程式，已經操作過 Code.org 對 APP Inventor 操作不再感到害怕，覺得不是那麼的困難而有信心。研究也發現在「我覺得學習 APP Inventor 可引發學習程式設計的動機」填答不同意的，在「我希望以後有機會可學習更多的程式設計課程」也填答不同意，因此若能提升學生學習程式的動機，就會促使學生未來想再學其他程式設計課程。
2. 教學教材面：現在學生用手機的頻率很高，主要以玩遊戲為主，玩手機時間越長，越不喜歡動腦筋，不想學習。應日系較屬於手機重度使用者，應日系學生在所有學習成

效滿意度問卷題項的回答中呈現較多極端值較多，有少數同學呈現非常不同意的情形，因此在教師的教學評量應日系的滿意度均低於國貿系，在研究者進行的各項學習成效滿意度問卷也是呈現一樣的結果。有些應日系的學生高中職就讀資訊相關科系，喜歡看動漫，不想未來只會寫程式，故改唸應日系，反想到還是要學寫程式。因此，針對不同背景的學生及學習特質應在教材設計上給予不同的內容，故教師在授課前可先了解學生背景，依學生背景設計相對應的教材及教學案例才能吸引學生學習。

3. 學生學習面：學生在學習程式時，沒有真正理解程式的用法，自己練習操作時，因為手機使用方便，直接錄影或拍下老師的操作解答，照老師的教學步驟做，上課也沒有記筆記的習慣，課後的練習太少，等到考試時不會應用。因此，若要能設計一套合適的專屬教材學習筆記，並透過融入式課程設計的教學介入筆記學習策略，協助學生學習記憶、練習與複習，以提升學生學習成效。

(三) 學生學習回饋

針對期中成績不理想進行檢討，學生的自我檢討與回饋歸納如下：

- (1) 沒有真正理解程式用法，不會應用：

“平時上課老師教的我都能跟上，也有充足的時間幫助同學，但是考試時沒有老師的程式就做不出來，代表平常我只有做而沒有真正的去理解”、“有時根本不知道作業到底要怎麼做，需要靠著錄影及相片才能把作業做出來”、“在學校裡做作業都會做，但是如果自己回家做作業可能就不會做了”、“其實平常上課我真的很常聽不懂，或是聽到一半的就開始不懂，但實際考一次試發現有些東西自己能做得出來很感動，只是寫程式自己還是太弱了，真的要多努力”、“好好地把每次上課的東西複習，理解每個東西到底是甚麼意思，這樣才可以出甚麼題目，就知道怎麼解題”、“我覺得主要是我都是用背的，所以當遇到變化題時頭腦會轉不過去，所以下一次自己練習時，就要更努力先了解他的邏輯而不是死背”、“回家有空閒時間思考程式為何要這樣設計也許會有助於設計程式上的進步”、“覺得自己的問題比較大在操作程式上感覺需要懂得變通，不懂就算有相似的題目也不會操作”、“要常常練習並且理解為什麼怎麼做”、“多練習程式集的架構及思考，應該能更了解程式該怎麼做”、“考試有點太難了有些變化題，不知道怎麼做如果考試能用平常上課時教的題目或許同學門能針對上課的題目去做練習”、“平常做作業的時候基本上沒有太大問題，但有時候會有不懂為什麼程式那樣寫，一知半解”、“真正了解程式設計的內容，遇到題目時會比較融會貫通”、“不是交作業就完成而是真正的去理解，不然一到考試時也是做不出來”、“加強邏輯。課程中想想看自己有沒有理解每個步驟”、“有一些是之前教過類似的，發現自己不會變通，在學習時應該要多方面的去理解它的原理及其應用”。

- (2) 課後的練習太少：

“聽的懂老師講的，可是實作的時候要看拍照下來的東西才能做出來，代表不熟悉，可能要再多花一點時間熟悉”、“經過這次的段考我發現需要再提早一點準備才能有更充足的時間學習還有上課的內容要回家溫習才不容易忘記”、“花多一點時間準備，老師上課很好理解但回家後還是要複習畢竟書是要自己看的”、“自己在這堂課的學習除

了平常作業以外並沒有花太多時間在這科上面，希望自己下次段考一定要更加努力”、“我沒拿到分的部分是程式設計，可能還要再多多練習，讓自己再更熟悉裡面的程式”、“自己的邏輯不好，可能需要多做幾次練習”、“考試之前在多練習多操作”。

(3) 不好的學習經驗、畏懼學習電腦：

“其實我真的對電腦不太擅長，每次上課我也都會盡力去聽、去懂但是我真的還是對寫程式這塊沒辦法”、“高中的時候有學過一段時間的 APP Inventor 可是高中老師講的超級不清楚，完全聽不懂之後就很討厭。上大學一開始學的時候也有點排斥，但因為老師很努力的講解，我也漸漸的有了一點興趣，想要學好這個，雖然還是不太熟悉，不過會繼續努力學習的”、“這堂課程讓我學到了平常沒有接觸過的東西，我覺得程式設計是個很複雜的東西，但是透過方塊拼湊的方式其實蠻有意思的，雖然我對這方面不是很上手”、“老師的教學很清楚了 但是電腦這個東西真的太複雜了..”

(4) 上課沒有記筆記：

“希望可以有一些筆記，確定哪一種題目要用哪一種程式”、“將更加詳細的程式過程的製作過程跟原因弄成圖片或筆記可以更方便了解跟複習”、“多練習提升熟練度上課要多寫筆記”、“搭配課本一起教學，讓我們能夠在課本上畫重點、做筆記、貼 memo 寫補充，這樣考試開書看也比較好找相關的內容”

多數學生反應期中考試題目 80%以上的同學覺得有點難，70%以上學生覺得期中題目內容與作業相似度差異在 20%上，11%學生不知道要參考什麼所以完全沒幫助，10%學生覺得題目與作業都不一樣無法參考。針對學生的期中考自我檢討上，很多同學提到老師出的題目與平時作業相似度差異太大，雖然 Open Book 考試但覺得題目與作業都不一樣無法參考及不知道要參考什麼所以完全沒幫助，歸究其原因，因為在做平時作業時只照老師步驟操作但都沒有真正理解及思考為何要這樣做，所以考試時不會應用。

“平時上課老師教的我都能跟上，也有充足的時間幫助同學，但是考試時沒有老師的程式就做不出來，代表平常我只有做而沒有真正的去理解”、“有時根本不知道作業到底要怎麼做需要靠著錄影及相片才能把作業做出來”、“有時候會不太了解為什麼是放這個”、“好好地每次上課的東西複習，理解每個東西到底是甚麼意思，這樣才可以出甚麼題目，就知道怎麼解題”、“我覺得主要是我都是用背的，所以當遇到變化題時頭腦會轉不過去，所以下一次自己練習時，就要更努力先了解他的邏輯而不是死背”、“回家有空閒時間思考程式為何要這樣設計也許會有助於設計程式上的進步”、“覺得自己的問題比較大在操作程式上感覺需要懂得變通，不懂就算有相似的題目也不會操作”、“需要再更熟悉程式邏輯”、“要常常練習並且理解為什麼怎麼做”、“多練習程式集的架構及思考，應該能更了解程式該怎麼做”

六. 建議與省思

經過本計畫之執行發現教務行政政策如編班方式會影響任課老師的教學成效，國貿系以資管群學生較多，應用外語系以外語及其他群學生較多，在高中職有學程式設計國貿系占 67.31%，應日系的學生沒學過多數占 60.71%，有學過僅有 39.29%，交叉比對後發現，

國貿系高中職有增 5 位學過程式設計相關課程，應日系的學生僅增加 1 位。亦即，在這些大一新生沒學過程式設計占 47.22%。國貿系與應日系編班方式不同，國貿系依特性進行分班，相同背景學生儘量編在同一班，對於教師專業教學上，學生比較不會產生落差太大的狀況，而應日系則是各種管道的學分平均分班，造成少部份學生學習上適應較為困難，有因此建議若能改變編班方式將相同背景編在同一班有利於教師在教學內容的安排及掌握學生學習狀況上。

本研究將學生高中職背景分為商管群、資管群及語文與其他群類，資管群的學生在完成 Code.org 程式感到很有成就感及認為學習 Code.org 可加強程式觀念的理解的平均數均低於其他兩群，這與一般認知資管的學生在學習程式上應該比較容易些有所不同，可能是 Code.org 較為簡單，一般沒有學過程式的同學，都很容易學習，包含國中小學生，故對資管的學生而言，因已學習過程式反而較沒有成就感及不覺得對程式觀念的理解有幫助。

經卡方檢定發現高中職背景在喜歡很有挑戰性的事物，以外語及其他群 60.8% 學生最高，其次為資管群 46.9%，商管群 20.0% 最低。學習背景在三種軟體學習成效上也呈現顯著差異，不論在 Code.org、Flowgorithm、APP Inventor 學習成效上，資管群最高，外語及其他群類次之，商管群最低。在學習 Flowgorithm 感到緊張，商管群最高占 28.0%，外語及其他群類次之占 25.5%，資管群最低占 6.3%。看起來商管群的學生在學習 Flowgorithm 時最容易感到緊張。由此看出商管群最不喜歡挑戰性的事物，在學習程式的滿意度也最低，這是一個很有趣值得探究的議題，若未來有機會可建議做進一步的探討其原因。

未來課程設計之建議：外貿學院學生在程式設計課程中自主學習能力差，故「學生自我學習狀況」及「期待本課程開設意願」與其他專業課程相較呈現較低滿意度，故設計一套好的教案，提供系統性的授課方法，配合課程內容搭配適性的教案教法及循序漸進的教學任務，得以提高學生學習動機，引導學生無懼跨領域學習，以培養學生解決問題能力及提升學生程式力及學生學習成效。然根據前期研究教學現場之問題從學生期中考回饋發現：設計符合技職校院非資訊科系學生程式力專屬學習教材筆記本協助學生學習增加記憶思考，提高練習與複習的次數是有必要的，原因如下：

“希望可以有一些筆記，確定哪一種題目要用哪一種程式”、“將更加詳細的程式過程的製作過程跟原因弄成圖片或筆記可以更方便了解跟複習”、“多練習提升熟練度上課要多寫筆記”、“搭配課本一起教學，讓我們能夠在課本上畫重點、做筆記、貼 memo 寫補充，這樣考試開書看也比較好找相關的內容”、“好好地把每次上課的東西複習，理解每個東西到底是甚麼意思，這樣才可以出甚麼題目，就知道怎麼解題”

基此，除了有完善的教學策略外，仍需有一套符合技職體系非資訊科系學習程式力之合適的教材，以及配合教材設計出相對應的專屬筆記本，記錄學生學習過程、訓練學生思考方式並能活用。一方面訓練學生專注力，利用學習五到（眼到、耳到、口到、心到、手到聯合運用），提升學生學習成效，二方面可深入瞭解學生學習質化差異，做為教師改善教學之基礎。

貳、參考文獻

- (1) Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20-23.
- (2) Berrett, D. (2012). How 'flipping' the classroom can improve the traditional lecture. *The chronicle of higher education*, 12(19), 1-3.
- (3) Caspersen, M. E., & Bennedsen, J. (2007). *Instructional design of a programming course: a learning theoretic approach*. Paper presented at the Proceedings of the third international workshop on Computing education research.
- (4) Cooper, S., & Dann, W. (2015). Programming: a key component of computational thinking in CS courses for non-majors. *ACM Inroads*, 6(1), 50–54. doi:10.1145/2723169
- (5) Frydenberg, M. (2013). Flipping excel. *Information Systems Education Journal*, 11(1), 63.
- (6) Gomes, A., & Mendes, A. (2007). *Learning to program - difficulties and solutions*.
- (7) Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, 42, 38-43. doi:10.3102/0013189X12463051
- (8) Hmelo, C. E., Gotterer, G. S., & Bransford, J. D. (1997). A theory-driven approach to assessing the cognitive effects of PBL. *Instructional science*, 25(6), 387-408.
- (9) Keller, J. M. (1987). Strategies for stimulating the motivation to learn. *Performance and instruction*, 26(8), 1-7.
- (10) Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. *The Internet and higher education*, 8(1), 13-24.
- (11) Mayring, P. (2000). Qualitative Content Analysis. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research [On-line Journal]*, <http://qualitative-research.net/fqs/fqs-e/2-00inhalt-e.htm>, 1.
- (12) McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., . . . Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In *Working group reports from ITiCSE on Innovation and technology in computer science education* (pp. 125-180).
- (13) Prensky, M. (2003). Digital game-based learning. *Computers in Entertainment (CIE)*, 1(1), 21-21.
- (14) Reigeluth, C. M. (2013). *Instructional design theories and models: An overview of their current status*: Routledge.
- (15) Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1),

68.

- (16) Salinger, S., Plonka, L., & Prechelt, L. (2008). A coding scheme development methodology using grounded theory for qualitative analysis of pair programming. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*.
- (17) Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- (18) Wager, W. ,”闡明學習成效－傳達您對學生的期望：臺灣高等教育部分主題探討”，*通識教育季刊*，第10期，第3-4卷，2003：頁101-114。
- (19) Young, D. J. (1991). Creating a low-anxiety classroom environment: What does language anxiety research suggest? *The modern language journal*, 75(4), 426-439.
- (20) 何昱穎, 張智凱, & 劉寶鈞. (2010). 程式設計課程之學習焦慮降低與學習動機維持-以 Scratch 為補救教學工具. *數位學習科技期刊*, 2(1), 11-32.
- (21) 李亦君. (2019). 運用鷹架學習、 創意自我效能與創造力於人文設計領域程式設計課程的實踐.
- (22) 林育慈, & 吳正己. (2016). 運算思維與中小學資訊科技課程. *教育脈動*, (6), 5-20.
- (23) 黃政傑. (2014). 安置性教室的理念, 問題與展望. *臺灣教育評論月刊*, 3(12), 161-186.
- (24) 蔡政宏. (2019). 應用邏輯式創新教學對學習效果影響之研究. *全球商業經營管理學報*(11), 45-56.
- (25) 溫子新. Rubric 評分表編製與使用 ,
http://aca2.tmu.edu.tw/files/recruit/42_c69b73be.pdf at 2020/12/04.
- (26) 溫廷宇.(2011). 網路學習成效之影響因素探討. *工業科技教育學刊*, 4 期, P41.

參、附件

附件一、學生基本資料問卷題目

基本資料	
Q1	性別
Q2	就讀的科系
Q3	高中職的科系
Q4	入學管道
Q5	有屬於自己的電腦
Q6	平常每天使用電腦或平板電腦的總時間?
Q7	平常每天使用手機玩遊戲時間?
Q8	平常每天使用手機上網時間?
Q9	是否曾使用/學習過程式設計工具
Q9.1	曾使用過的程式設計工具為何:(可複選)
Q10	高中職是否有程式設計相關課程
Q10.1	高中職的程式設計相關課程在幾年級

附件二、學習前自我認知問卷題目

構面	編號	題目
自我期許與 認知	R1	相較於班上的其他同學，我認為自己是個好學生。
	R2	相較於班上的其他同學，我預期我會有好的表現。
	R5	我會盡力完成教師要求的作業。
	R8	我喜歡很有挑戰性的事物。
學習態度	R3	遇到較困難的學習內容時，我通常會放棄或只學習比較簡單的部分。
	R4	上課時我很容易看其他科目的書或玩手機。
資訊科技接 受度	R6	我覺得資訊科技對生活有很大的影響，例如：智慧型手機、物聯網、人工智慧、機器人、大數據…等。
	R7	我覺得在未來有很多目前的工作被機器人或人工智慧取代。
程式力期待 與認知	R9	我覺得將程式設計相關課程內容學好是重要的。
	R10	我覺得程式設計很好玩。
	R11	我對寫程式很有興趣。
	R13	我希望透過此課程學習更多程式相關知識。
	R14	尚未修此課程前，我很期待本課程的開設。

附件三、Code.org 學習成效問卷題目

構面	編號	題目
學習動機	C15	學習 Code.org 程式設計讓我覺得學習程式設計很有挑戰性。
	C16	給我題目，我就可以自己想出解題步驟(不需要老師提示)。

	C17	我覺得學習 Code.org 可引發我學習程式設計的動機。
	C20	我覺得完成 Code.org 程式很有成就感。
學習興趣	C1	我覺得 Code.org 是一個很好玩的軟體。
	C2	我覺得學習 Code.org 程式設計令我覺得很自在。
	C6	我覺得 Code.org 程式設計操作過程很簡單。
	C9	我覺得學習 Code.org 程式設計很輕鬆。
	C10	我覺得 Code.org 程式設計課程很有趣
使用介面及環境	C7	我覺得 Code.org 程式設計並不會很抽象。
	C8	我覺得 Code.org 程式設計很容易理解。
	C11	我覺得 Code.org 軟體的使用者介面環境很方便使用。
	C13	在學習 Code.org 時有時會看不懂題目要做什麼
	C18	我覺得 Code.org 軟體之拼塊式(拼圖式)的使用，方便我學習程式設計。
學習態度	C3	我覺得可以容易適應學習 Code.org 程式設計。
	C4	我覺得學習 Code.org 時，總覺得時間過得很快。
	C5	我覺得學習 Code.org 不會讓我感到緊張
	C12	我在 Code.org 程式設計課程有認真學習。
	C24	我希望以後有機會可學習更多的程式設計課程。
學習態度	C14	學習 Code.org 程式設計後讓我覺得學習程式設計並沒有想像中的難。
	C19	操作拼塊的過程中能夠幫助我將想法視覺化。
	C21	我覺得學習 Code.org 可加強我對程式觀念的理解。
	C22	我喜歡將我的作品分享出來，讓別人來觀摩。
	C23	我以後還會繼續使用 Code.org 軟體來創作作品
	C25	整體而言，我滿意 Code.org 程式設計課程的學習

附件四、Flowgorithm 學習成效問卷題目

構面	編號	題目
學習動機	F15	學習 Flowgorithm 程式設計讓我覺得學習程式設計很有挑戰性。
	F16	給我題目，我就可以自己想出解題步驟(不需要老師提示)。
	F17	我覺得學習 Flowgorithm 可引發我學習程式設計的動機。
	F20	我覺得完成 Flowgorithm 作業很有成就感。
學習興趣	F1	我覺得 Flowgorithm 是一個很好玩的軟體。
	F2	我覺得學習 Flowgorithm 令我覺得很自在。
	F6	我覺得 Flowgorithm 操作過程很簡單。
	F9	我覺得學習 Flowgorithm 很輕鬆。
	F10	我覺得 Flowgorithm 課程很有趣。
	F7	我覺得 Flowgorithm 程式設計並不會很抽象。

使用介面及環境	F8	我覺得 Flowgorithm 程式設計很容易理解。
	F11	我覺得 Flowgorithm 軟體的使用者介面環境很方便使用。
學習態度	F3	我覺得可以容易適應學習 Flowgorithm。
	F4	我覺得學習 Flowgorithm 時，總覺得時間過得很快。
	F5	我覺得學習 Flowgorithm 不會讓我感到緊張
	F12	我在 Flowgorithm 課程有認真學習。
	F23	我希望以後有機會可學習更多的程式設計課程。
學習成效	F13	我覺得學習 Flowgorithm 可加強我對程式觀念的理解。
	F14	學習 Flowgorithm 讓我覺得比較理解電腦的運作。
	F18	Flowgorithm 的流程圖有助於理解要表達的是什麼。
	F19	操作 Flowgorithm 過程中能夠幫助我將想法視覺化。
	F21	未來遇到比較難解的問題，我會繼續使用 Flowgorithm 來釐清問題。
	F22	我喜歡將我的 Flowgorithm 作品分享出來，讓別人來觀摩。
	F24	整體而言，我同意 Flowgorithm 程式設計課程的學習

附件五、APP Inventor 學習成效問卷題目

構面	編號	題目
學習動機	A10	我覺得使用 App Inventor 設計出程式打包成 apk 至手機執行，給我很大的成就感。
	A21	我覺得學習 APP Inventor 可引發我學習程式設計的動機
學習興趣	A1	我覺得利用 APP Inventor 設計實用的 APP 軟體，很有趣
	A4	我覺得 APP Inventor 程式設計操作過程很簡單
使用介面及環境	A5	我覺得 APP Inventor 軟體的使用者介面環境很方便使用
	A6	我覺得 APP Inventor 程式設計很容易理解
	A8	我覺得透過 Google 登入 MIT 雲端平台執行 App Inventor 軟體很方便
	A9	我覺得利用夜神模擬器(NOX)執行 App Inventor 設計好的程式很方便和穩定
	A11	我覺得電腦教室軟體設備的穩定性，會影響我 App Inventor 程式設計的學習
3. 學習態度	A12	我覺得 App Inventor 拚塊式(拼圖式)的使用方式，讓我覺得學習程式設計很簡單
	A2	我覺得可以容易適應學習 APP Inventor 程式設計
	A3	我覺得學習 APP Inventor 時，總覺得時間過得很快
	A7	我有花時間在 App Inventor 程式設計的課後學習
	A23	我在 APP Inventor 程式設計課程有認真學習
學習成效	A24	我希望以後有機會可學習更多的程式設計課程
	A17	學習 APP Inventor 程式設計後讓我覺得學習程式設計並沒有想像中的難
	A18	我覺得 App Inventor 課程之學習可以培養我發現問題與解決問題的能力

	A19	我覺得 App Inventor 課程之學習可以訓練我邏輯推演的能力
	A20	我覺得 App Inventor 程式設計創意應用發想企畫書，可以培養我跨領域的思考能力
	A22	我覺得 APP Inventor 證照考試很簡單，讓我不害怕學習程式設計
教學內容 與方法	A13	我覺得 App Inventor 課程範例的學習，有利於我學習程式設計
	A14	我覺得 App Inventor 課程安排順序，可以幫助我的學習
	A15	我覺得 App Inventor 課程內容，不會太難
	A16	我覺得 App Inventor 課程內容，進度不會太快
	A25	整體而言，我很同意 APP Inventor 程式設計課程的學習

附件六、基本資料卡方檢定彙整表

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q10.1
Q1											
Q2											
Q3		.000									
Q4	0.004										
Q5											
Q6	0.005										
Q7	0.003										
Q8	.000						.000				
Q9		0.036	.000								
Q10		0.004	.000				0.036		.000		
Q10.1		0.032	.000			0.013			.000	.000	

附件七、基本資料與學習自我認知卡方檢定彙整表

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q10.1
R1	0.031					0.003					
R2											
R3		0.015					0.023	0.016	0.045		
R4											
R5	0.002						0.016		0.038		
R6			0.003								
R7							0.011				
R8			0.04								
R9											
R10	0.018						0.004	0.021			
R11											
R12											

R13				0.045			0.025				
R14											

附件八、基本資料與 Code.org 學習成效卡方檢定彙整表

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q10.1
C1									0.029	0.004	
C2							0.039				
C3		0.032								0.047	
C4											
C5		0.021									
C6	0.01		0.038						0.003		
C7		0.009									
C8											
C9											
C10											
C11						0.042					
C12											
C13									0.013		
C14								0.045			
C15										0.046	0.02
C16	0.022							0.026	0.01		
C17		0.024									
C18											
C19							0.01				
C20											
C21											
C22											
C23											
C24											
C25											

附件九、基本資料與 Flowgorithm 學習成效卡方檢定彙整表

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q10.1
F1			0.028							0.02	
F2		0.013	0.041							0.009	
F3											
F4		0.048					0.005				

F5			0.038								
F6										0.034	
F7											
F8		0.024	0.005						0.018	0.01	
F9										0.012	
F10									0.036	0.02	
F11											
F12									0.026		
F13											
F14											
F15											
F16	0.036										
F17											
F18											
F19							0.03				
F20											
F21											
F22						0.011					
F23						0.002				0.031	
F24											

附件十、基本資料與 APP Inventor 學習成效卡方檢定彙整表

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q10.1
A1		.007							.026		
A2									.013		
A3		.008									
A4					.000					.018	
A5		.008									
A6		.008	.038		.000				.022	.014	
A7		.004									
A8											
A9											
A10		.000							.049		
A11			.025								
A12		.010	.022						.007	.027	
A13		.036		.009							
A14		.013								.030	.010
A15		.026						.027	.004		

A16								.028			
A17							.041	.014			
A18											
A19					.000				.044		
A20		.012	.013		.007					.015	
A21		.044			.018					.028	
A22									.001		
A23											
A24					.018				.021	.005	.047
A25			.015						.013	.005	