

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：PEE1080029

學門專案分類/Division：工程

執行期間/Funding Period：2019/08/01 ~ 2020/07/31

運用 **ADRI** 教學模式結合成對編程與線上評測於程式設計課程
之行動研究

(配合課程：基礎程式設計、進階程式設計)

計畫主持人(Principal Investigator)：

許育嘉

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

國立臺灣體育運動大學 運動資訊與傳播學系

成果報告公開日期：

立即公開 延後公開(統一於 2022 年 9 月 30 日公開)

繳交報告日期(Report Submission Date)：2020/09/11

運用 ADRI 教學模式結合成對編程與線上評測於程式設計課程之行動研究

1. 研究動機與目的

「程式設計」對於學生而言，是一門挑戰性極高的課程。目前臺灣的大學入學新生，極大多數都是屬於程式設計的初學者，普遍都是在進入大學後，才開始接觸到程式語言。儘管有少部分學校在 12 年國教階段，已開始導入運算思維、創客自造、積木式程式設計、機器人等之基礎程式設計教學，但各個學生因為國高中階段所接受到的程式運算、邏輯訓練、問題解決等課程差異極大，因此在教授程式設計課程的教師，也必須同時考量到學生數位落差的問題。

近年來在順應世界程式教育的潮流，並深植發展人工智慧的基礎實力，教育部新的 108 學年度十二年國民基本教育課程綱要，也已納入「程式設計」領域課程，程式設計的課程，在邁向人工智慧時代的過程中，越來越顯得重要。然而根據相關研究文獻指出，程式設計在世界各國大學實施的狀況，不管是什麼科系背景的學生，學生學習的情況大致相同，大都維持著低通過率、重修人數比例高、學習動機低落等問題。而在研究者於所任教的程式設計課程中，多年下來也觀察到相似的現象。

Mederiros, Ramalho,和 Falcão (2018)為了探討程式設計課程學生學習成就低落的問題，採用文獻回顧法，系統化地經由檢索、探索引文等方式，篩選出 100 篇的研究文獻進行分析，來解答三個研究問題：程式初學者需具備哪些關鍵學前技能與背景知識？初學程式的學生遇到那些學習困難？教師課堂上的挑戰為何？研究結果指出，要學好程式設計需具備的相關基礎技能，依序為：問題解決能力、數學能力、學過程式設計運算思維、摘要能力。而學生學習的困難可歸納成：問題的界定與形成、解決方案表達、解決方案的執行與評估、學習行為四大類，而各類中最常見的困難分別是問題解答、程式語言語法、除錯技巧、動機與投入。而教師所面臨的諸多挑戰，最常見的幾個依序為：教學方法與工具、維持學生的動機投入和堅持、師生的溝通與回饋。

同樣對於學生在程式設計課程的通過率不高，提升學習成效的問題上，也有學者針對所採用之教學介入的形式，以文獻分析法進行系統化的回顧(Vihavainen, Airaksinen, & Watson, 2014)。相關程式設計教學的文獻，經過檢索、過濾、篩選後，剩下 32 篇來分析教學介入的形式，將教學介入的形式，以詞彙方式來編碼，之後進行統計分析。最後歸納出，有下列十種常見的活動形式，依頻率最常見的依序為：教材變革、同儕支持、協同合作、情境化、配分比重、媒體運算、遊戲主題、協助支援、先修課程、團隊合作學習。而其中有些是採取多種教學介入之混合方式，從教材建構、工具的使用、評分比重的調整、成對編程、團隊專案製作等，對傳統課程的教學模式進行改革，可提供學生課程的通過率。

程式設計的能力，就如同過去所謂數學為科學之母般，是未來資訊科技應用發展的重要關鍵，越來越受重視。舉凡在創新行動 App 開發、新型態的網路社群行銷、大數據分析、人工智慧應用，如能具備程式設計的技能，不論是在哪個產業皆能善用資訊科技，創造知識經濟的產值，厚植資訊國力。因此本研究旨在探討程式設計課程，如何在教學時數不足、學習動機不高下，引入新的教學方法，設計更符合參與課程學生特性的教材，以提升學生的學習成效。程式設計教學的目的，並不是要讓人人都能成為專業的程式設計師，也不見得在各行各業裏都須要專業的程式設計師，但是當遇到能用程式輕易解決問題時，能有使用程式解決問題的能力。

2. 文獻探討

(1) ADRI 教學模式

程序部署結果改進模式(Approach, Deployed, Result, Improvement, ADRI)為一種廣被採用品質確保模型，最早是由 Walter Shewhart 以 Plan-Do-Check-Act(PDCA)模型為基礎，於 1939 年發展出來的(Johnson, 2016)，之後也經常被延伸應用到教育和商業領域中(Razvi, Trevor-Roper, Goodliffe, Al-Habsi, & Al-Rawahi, 2012)。在商業上，澳洲籍紐西蘭的企業評比機構，採用 ADRI 模型來對企業進行品質獎的評選，在教育產業中，澳洲與紐西蘭的大學，以及臺灣的大學評鑑中，也採用了 ADRI 模型來建立教學品質的保證機制。

ADRI 在程式設計的教學應用，最早是 Malik 等學者應用 ADRI 模型發展創新的系統化教學模式，於基礎程式設計課程中實施，以及用來提升學生的學習成效，並進行一系列的延續性研究(Malik & Harsh, 2013; Malik & Coldwell-Neilson, 2017a; Malik & Coldwell-Neilson, 2017b); Malik, 2018)。這些研究結果，均能反映以 ADRI 的程序進行程式設計教學，可以獲得學生正向的回應，並提高了修課的通過率，同時教師也可由研究中來分析所採用的教材和評量工具，調整教學的方式。

ADRI 模型的教學建議，主張程式初學者在課程學習的過程中，應該同樣地重視解答問題的策略和程式語法語意的知識上，透過 ADRI 模型的四個階段設計，可以實踐上述目標。四個階段的程式教學設計如圖 1 所示，說明如下：

程序(Approach)：著重於問題解答策略，例如運用 pseudo-code 和流程圖，針對問題敘述提出解答。學生了解問題敘述與限制後，可提出一般解和演算法，而不需考量是否具備程式句法和語意的知識。

部署(Deployment)：著重於程式語言的句法和語意，了解程式語言的規則和結構。學生須將前一階段提出的演算法，開發為程式。並且了解直譯、編譯的過程，以及測試和除錯的技巧。

結果(Result)：著重於程式的輸入、輸出、處理程序，以及共通的程式撰寫錯誤。學生必須了解解答問題的程式，了解程式輸出輸入格式的規定和資料型態，一些程式共同慣用的語法，以及了解當程式邏輯錯誤時會如何呈現。

改進(Improvement)：於原本的問題上，增加新的問題描述與要求，用來增加實作的經驗，並且嘗試以另外一種程式語言結構，來解答同樣的問題。學生須著重於分析新增的需求，並且這些需求以原先的程式進行擴充，提供更深入的程式實作機會。

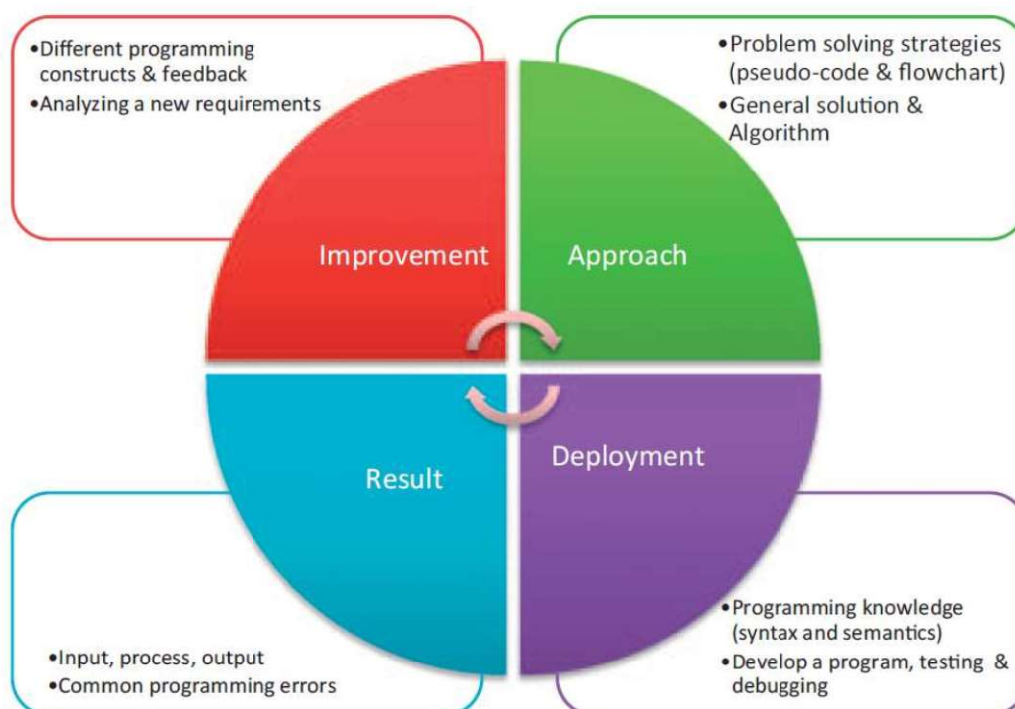


圖 1、ADRI 四階段的程式教學模式(資料來源：Malik & Coldwell-Neilson, 2017b)

(2) 自動化程式線上評測(online judge)

以程式設計而言，了解程式設計的原理，與了解如何寫出程式，仍有一段距離。以程式的撰寫而言，一開始都是接觸到片段的幾行程式碼，範圍大約是該次上課的章節。當需要利用寫程式來解決問題時，通常都是到了下半個學期，教學進度累積到一定程度時，而學生此時才會深深體會到，由於先前的片段程式碼基礎學的不夠好，而無法利用程式來解答問題。因此有研究者提倡(Brito & De Sá-Soares, 2014)，基於建構式教學主義的理念，知識的獲得來自自身經驗與體驗，而不是經由某種類型的資訊傳遞，因此在程式實作能力的提升上，應當增加學習成效評量的頻率，並結合系統化的教學模式。評量的頻率最好是一週一次，因為一週的時間尚可讓學生去改變學習的方法，並且能讓教師能適時提醒那些需要關注的學生。而當要實施每週評量時，沒有額外的資源很難達成，此時自動化的評測系統可做為程式設計教學輔助的選擇。

程式實作能力的提升，可藉由具自動化程式線上評測功能之教學輔助平台的導入，讓學生有更多實作練習的機會。傳統程式上機實作的教學流程，是在課堂上實施，在教師講述該課程教學內容與範例後，安排實例讓學生實際練習程式編碼，再輔以課堂的答疑與實驗指導。但受限於課堂時間及生師比的問題，課堂實作往往變成了打字練習，練習輸入程式碼。而一位教師要面對全班數十位的學生，很難在課堂上個別一一去解答學生的問題。而不論是課堂上的實作，或是安排課後作業實作，學生是否有確實進行實作，最大的關鍵仍在於主動性。

自動化程式線上評測系統，在許多教學研究中被提出作為輔助教學的工具。教師可在教學平台上，依據課程的內容設計程式實作的題目，學生則可於平台上作答，實作編碼或提交已完成的程式碼，平台會自動依據作答自動進行語法剖析與輸入輸出測試，給予分數。並記錄學生的學習次數、分析各題目之答對狀態、學生學習的弱點等。

Gou, Wen, & Liu (2017)引入了線上PTA(Programming Teaching Assistance)教學在線判題系統，實施於兩個計算機類科系的大一班級，每班均為40人，於程式設計基礎課程中教授C語言。其中一個班級採傳統的程式實作練習，另一個班級採新型態的線上線下程式實作練習。研究結果顯示，採用新型態的實作練習，在期末考的表現上，學生獨立完成題組作答的能力有很大的提升，而平均答題與練習數量也有顯著的提升。

而不僅是線上評測，進一步也有學者以自動線上評測的平台為基礎，進一步發展自適性的程式設計學習平台(紹陽林，少桓董，宗德林，筱倩曾，2017)，平台會依據學生目前實作練習的評分結果，依學生程度自動分派下一個練習的題目，針對不熟、不懂、答錯的部分加強練習。

而除了線上評測、自適性學習的功能外，有的平台更加強化程式實作時，在答題及更正錯誤時的立即回饋功能(Palominos, Palominos, Durán, Córdova, & Díaz, 2018)。傳統課堂上進程式實作時，當學生有編碼的問題時，教師須到學生電腦前，一起看著螢幕進程式碼除錯。而透過平台學生直接反映實作時的困難，而教師可以透過平台查閱程式碼與錯誤訊息，快速的了解學生的問題，給予指導，強化了傳統課堂上，師生溝通反饋的機制。

本研究為了更容易掌握，學生是否確實進行課後作業的練習，了解學生的學習困難，故將導入自動化程式線上評測平台，並設計合適的教學內容，搭配課堂實作與課後練習的增加，期望能提升學生的能力與學習成效

3. 研究問題

本研究所要解決之教學現場之問題，依上述章節之討論進行整理，主要為(1)學習動機不足(2)課程時數有限(3)符合學生特性的教材，我們將以ADRI之教學模式，來規劃教材與教法，並探討學生之學習回饋。

4. 研究設計與方法

(1) 研究架構

本研究採用行動研究(action research)模式來進行，在引進創新教學教法和重新設計教材的同時，執行經過規劃的教學行動方案、持續改進實務行動、監控並評鑑行動的實施歷程與結果，過程中透過反覆的循環，來改進教學的品質，提升學生學習的成效。

本研究實施的課程為程式設計，程式語言將選用 Python，可降低程式語法和語意的難度，並有利於未來人工智慧與大數據分析相關課程之接軌。在學生核心能力的部分，所設定的教學目標為：一、具備團隊合作精神，利用運算思維分析、解決問題之能力；二、熟悉程式設計之語法與語意，具備獨立程式編碼能力；三、解決實際問題的程式設計能力。而為了提升學生學習成效，針對上述之教學目標，擬訂教學改善之活動。

整體研究架構設計，如圖 2 所示，共包含四個部分：教學內容設計、課堂活動、合作學習與回饋反思，如圖六所示。第一部分為教師依 ADRI 模式，將解答問題能力與程式語法語意兩者並重之教學策略，融合在一起規劃教學內容。第二部分課程實作活動，為配合課程進度所實施之課堂程式實作及課後程式作業，利用教學平台提供自動評分之功能，讓學生能確實並多加練習，並透過平台之互動溝通功能，解答程式除錯的問題。第三部分合作學習希望透過 XP 的軟體開發模型成對編程的概念，採用合作學習的策略，讓學生兩個人一組相互激勵、一起腦力激盪、增加互動、共同解決問題。第四部份反思與回饋，透過教學反思日誌、學習回饋單、問卷與訪談的方式、歸納分析問題與建議，並對課程進行修正。

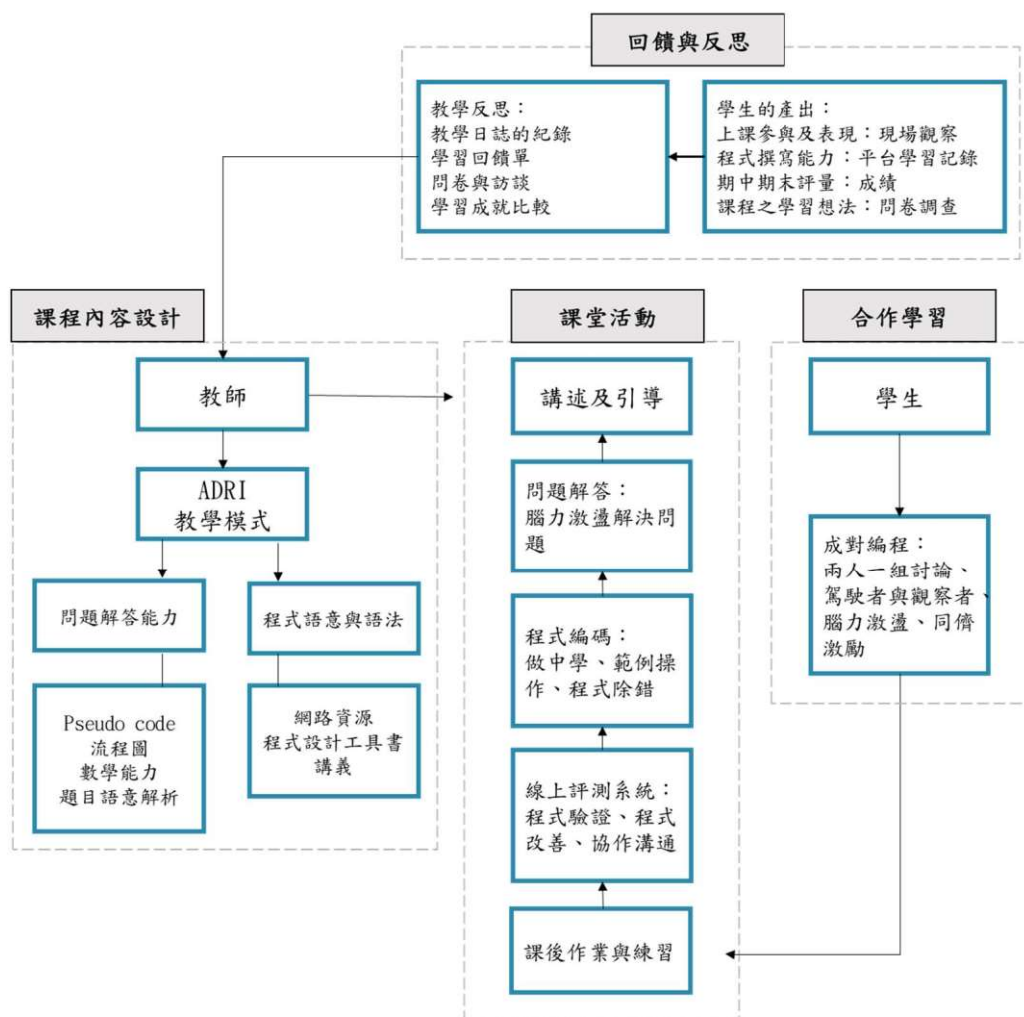


圖 2、研究架構

(2) 研究流程

本研究之流程如圖 3 所示，分述如下：

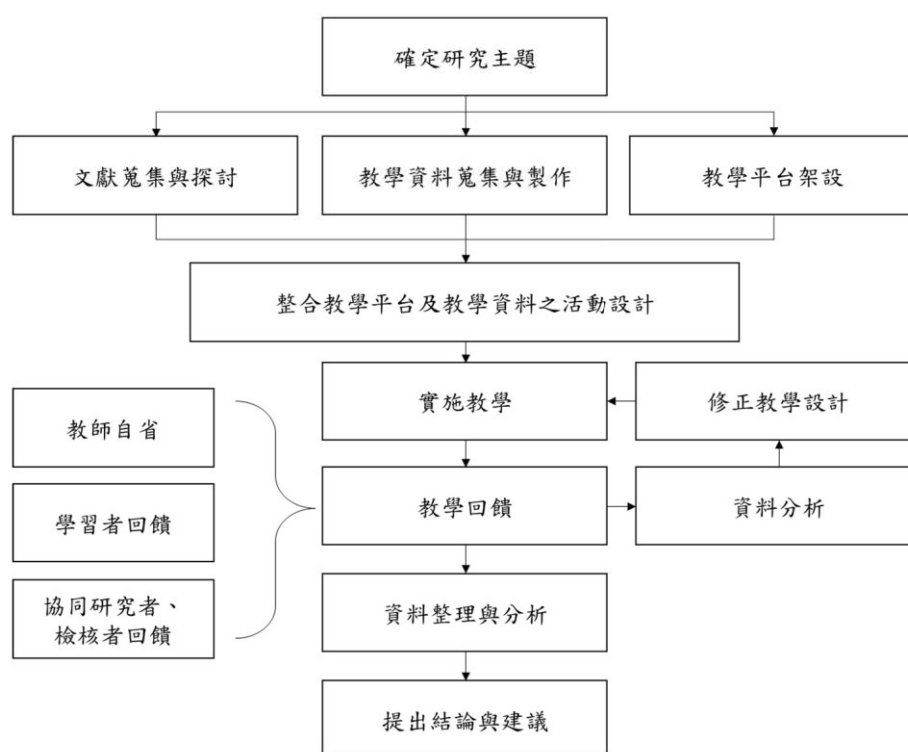


圖 3、研究流程

一、 確定研究主題

研究者由多年的實際程式設計教學經驗中，整理出幾個學習成效上的問題，做為研究的主題。

二、 文獻蒐集與探討

與程式設計教學相關的文獻，在構思研究主題階段已有初步的瞭解。而在執行計畫階段，相關問卷的製作、質性分析的結果、實施成效分析等，需再進一步的參考相關文獻進行探討。

三、 教學資料收集與教材編製

研究者依據教學主題與單元目標，廣泛蒐集程式設計的各種題目，由目前出版社所出版的程式設計書籍中，挑選與改編為適合研究對象於課堂實作與課後作業，並編製教學講義。本研究依 ADRI 模型，來發展程式設計教材，內容包括四個部分，首先讓學生以 psuedo-code 或流程圖為工具，來解決問題；其次是程式語言及語意的說明；接著讓學生依據程式的輸入、輸出要求進行編碼，並程式測試；最後，要求學生以其他的程式語法，重新解答同樣的問題。

四、 網路教學平台之設置

研究中將採用學校現有之 Moodle 平台，做為教材下載、問卷等教學輔助使用。此外，針對程式設計部分，則採用外部的工具及平台，來提供學習參考資源、自動化程式線上評測、程式碼標註與討論、生師互動等功能，目前已有 repl.it、code.org、Microsoft Azure Notebook、Code Judger、UVa Online Judge、URI Online Judge、Code Marshal Online Judge、Spoz Online Judge、GitHub 等平台，但各項功能支援的情形不一，需一一測試後選擇最合適的，或個別使用單一功能，做為本研究使用。當決定採用之教學工具平台後，將相關教材、範例程式碼等，建置於平台上。

五、 整合教學活動設計與教學平台

本研究在程式編碼時，將採用兩個人一組的成對編碼方式，來實踐合作學習的理念。而在教學實務上，分組實作、課後作業、專題開發之進行，需與教學平台的功能整合。

六、 實施教學

本研究預計實施教學的時間，配合計畫執行的時間一年，從 108 學年度第一學期開始，一直到學期結束，連續實施於兩個學期的連貫課程。約每 2 到 3 週為一個教學單元，每個單元均包含問題解決的教學、程式的語法及語意、課堂實作練習、課後程式作業。每階段進行一個單元的教學及作業，每階段完成後則進行反省與修正，再進入下一階段，如此循環。而在第九週及第十八週時，則為成就測驗的階段，實施期中與期末考試。於各階段教學結束後，再依據相關資料進行研究。

七、 教學回饋

教學過程中，蒐集學生的課堂練習作業、課後作業、回饋單、教學平台之互動內容、學生訪談記錄、問卷填寫結果、以及研究者之教學省思日誌等，做為教學回饋。

八、 修正教學設計

依據上一階段之教學回饋呈現之缺失或不足處，調整各部分講解與練習的時間、教材的難易度、練習與作業的頻率等，修正後於下一階段的教學活動實施。

九、 資料整理與分析

於每階段完成後之教學回饋資料進行彙整，待學期結束後，針對所蒐集的多方資料進行統計分析、統整，最後進行詮釋與分析。

十、 提出結論與建議

綜整兩個學期循環的資料，進行整理並歸納出結論，對結果提出相關建議。

(3) 研究對象

本研究以研究者所任教之程式設計課程修課學生為對象。課程為大一上學期屬於系定選修課之基礎程式設計，以及大一下學期屬於系訂分組必修課之進階程式設計，共兩個學期的課程。

(4) 研究方法及工具

本研究採行動研究法，在方法論上是採用折衷統合的方法，為質性與量化之綜合研究。其優點是可以運用許多不同的方法，蒐集資料。

在量化的部分，擬採用李克特式五分量表進行問卷調查，調查參與課程的學生對於教材內容各個知識點之學習困難，以及對教師課程進行方面之滿意度，以進行統計分析。而針對學習成效的部分，則以學習評量成績，進行統計分析。

而在質性的部分，則採用深度訪談法、學習回饋單，蒐集學生對於教學與自我學習成效之反饋意見，而對於教師則以觀察法，將教學省思記錄到教學日誌中，最後彙整多方資料進行分析，以概念圖、原因和影響分析、解釋性論述，來闡釋資料。

5. 教學暨研究成果

(1) 教學過程與成果

本研究配合兩個學期之課程，共分成四個階段實施，以每學期第九週期中考做為分界，說明如下表 1:

表 1、四階段之教學過程與內容

階段	期間	配合課程	修課人數	教材	研究活動
一	108 上 1~9 週	基礎程式設計	24	教科書:Python 初學特訓班(第三版)[基礎]	學習評量(期中考)
二	108 上 10~18 週			TQC+基礎程式設計題庫 偶數題、自動評測平台	學習評量(期末考)、問卷、自動評測平台使用統計
三	108 下 1~9 週	進階程式設計	20	TQC+基礎程式設計題庫 奇數題搭配流程圖、自動評測平台、認證考試	學習評量(期中考)、自動評測平台使用統計、質性訪談
四	108 下 10~18 週			選用教科書:Python 初學特訓班(第三版)[應用]	問卷

而在實施教學的第一週，我們針對學生之程式設計在大學入學前之學習經驗進行問卷調查，得知在修課的 24 位學生中，有 12 位的學生在大學入學前曾學習過程式設計，而學習的時間大多為一學期，學習的語言以 C/C++/C# 佔 5 位為最多，如圖 4 所示。

一、基本資料與背景：											
基本資料				畢業高中							
學號	姓名	班級	性別	縣市	高中/中學	過去是否「在學校」學過程式語言相關課程：	語言名稱	學了多久	過去是否「於課外」自己學過程式語言相關課程	語言名稱	學了多久
40409065	林	經濟四	男	臺中市	高中	否			否		
40509002	呂	經濟四	女	嘉義縣	高中	是	C/C++/C#	一學期	否		
40509012	呂	經濟四	男	台南市	二中	是			否		
40509062	呂	經濟四	女	阿緬省	附中	是	C++	一學年	否		
40509088	呂	經濟四	女	彰化市	高工	否			否		
40809010	吳	經濟一	男	彰化市	高中	否			否		
40809002	吳	經濟一	女	新北市	高中	否			否		
40809003	吳	經濟一	男	彰化市	高中	否			否		
40809014	吳	經濟一	女	基隆市	女中	是	Python	一學期	否		
40809005	吳	經濟一	男	彰化市	高中	否			否		
40809005	吳	經濟一	男	苗栗縣	高中	是	scratch	一學期	是	Java	一個月
40809008	吳	經濟一	男	彰化縣	高中	否			否		
40809019	吳	經濟一	女	彰化縣	高中	否			否		
40809020	吳	經濟一	女	彰化縣	高中	是	C/C++/C#	一學期	否		
40809022	吳	經濟一	男	台中市	高中	是	VB	一學期	否		
40809003	吳	經濟一	男	彰化市	高中	是	C#, APP Inventor	各一學期	否		
40809004	吳	經濟一	女	彰化市	女中	否			否		
40809035	吳	經濟一	男	高雄市	高中	是	C	上過N小時	否		
40809005	吳	經濟一	男	彰化縣	高中	是	VB	一學期	否		
40809009	吳	經濟一	女	彰化市	高商	是	Html	一學期	否		
40809002	吳	經濟一	男	台中市	一中	是	Java, Python	各一學期	否		
40809043	吳	經濟一	男	彰化市	高中	否			否		
40809044	吳	經濟一	男	嘉義縣	高中	否			否		
40809045	吳	經濟一	男	彰化市	附中	否			是	Java, C/C++/C#	

圖 4、學生對於程式設計之先前學習經驗

在教學階段二時，我們引入了 Code Judge 平台(如圖 5)，做為自動化程式線上評測系統，並搭配中華民國電腦技能基金會所開發的 TQC 程式設計的題庫，來進行教學。學生可以上課中及上課後，於平台中進行練習與評測，並會記錄練習的狀況。

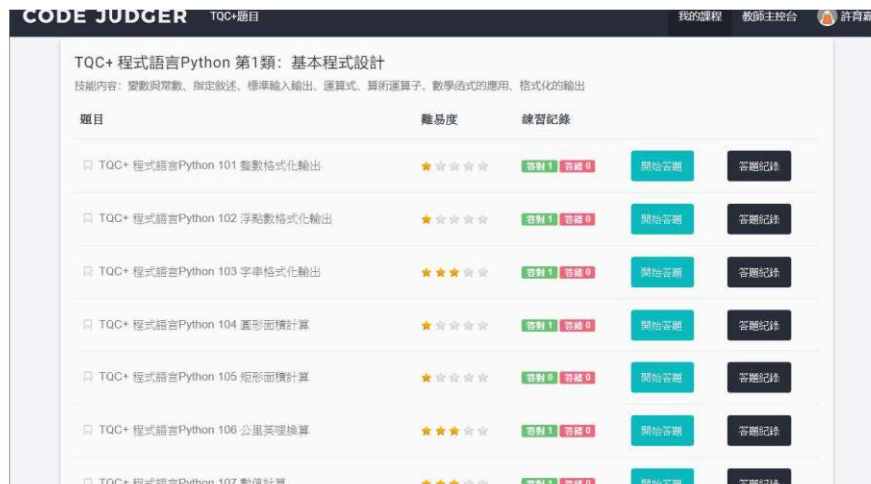


圖 5、Code Judger 平台中的 TQC 程式設計題庫

而在教學階段三，實施改良 ADRI 之教學模式時，我們引入了流程圖來加強解說邏輯判別式、迴圈等，學生反應比較難的教學單元，並以學校的 Moodle 教學平台(如圖 6)來輔助教學，將相關的流程圖、程式碼、及部分上課錄製的影音解說置放於教學平台上。



圖 6、輔助教學使用之 Moodle 平台

(2) 教師教學反思

在實施 ADRI 教學的同時，教師於課堂上透過與學生的互動中，也了解到學生的反應及回饋，記錄於教學省思日誌中，同時也在實施教學問卷調查同學的回饋意見後，對於全體的學生進行深度訪談(詳見附件一)，以了解學生學習的情況。經過教學實踐及諮詢同領域之專家意見(詳見附件二)後，研究者於過程中反思後，逐步改進教學，歸納發現如下：

一、採用程式自動評測系統題庫式的練習，能提升學習興趣與成就感

與傳統採用教科書之練習題相較，學生於程式自動評測平台上練習，更容易獲得成就感。主要是因為平台上的題目普遍較簡短也較容易，雖然題目比較多，但按部就

班一題接著一題練習，會比教科書上的題目，更有練習解題的意願。

二、改變考試的方式，易影響到學生學習評量的表現

採用真實的認證考試的上機考試環境，容易造成學生臨場緊張，容易失誤。而若採用與平時練習相同的環境進行上機考，則較無失誤發生。這點亦可從問卷中對於期中考的表現度，從第一學期 3.65 降到第二學期 3.35 的情況看出。

三、學生願意於課後花多少時間學習，影響學習成效甚鉅

雖然說學習動機與學習興趣，對於學習成效影響很大。但依研究中之質性訪談發現，學生的學習成效，與願意學習投入的時間影響更大。有學生提到對於程式設計有高的學習動機與興趣，且上課時也都非常認真學習和做練習，但在課後則因打工，反而於考前疏於練習，已至學習評量分數不高。

四、引入流程圖對於 ADRI 的教學模式有幫助

許多同學對於能引入流程圖，加強程式邏輯的講解十分認同。然而在質性訪談中，對於學習成就較差的學生，其回應為有些困難的題目，流程圖也是看不懂。

五、兩人一組的學習方式，接受度很高

本研究參考成對編程概念，於第四階段讓學生以兩人一組來進行學習，由問卷中得到學生極佳的回饋。

(3) 學生學習回饋

一、自動化程式線上評測系統使用情形

兩個學期的教學中，學生使用 Code Judger 平台之情形，全體學生類計次數如圖 7 所示。我們可以看到，在 108、309、310、409 這些題目上，正確率比較低，但是相對的這些題目，學生的答題次數也會比較多。這隱含的意義可解釋成，學生在遇到難題時，會盡力的去解決它、克服它，而不是放棄，這對於學習動機而言，呈現出正向。

而進一步我們對照學生之學習成就，將期中考與期末考成績與平台使用情形比對(如圖 8)，大致上可看出解題數、正確率與成績具有正相關。但因學生之樣本數(<30)不足，未進行統計相關係數計算及統計檢定驗證顯著性。

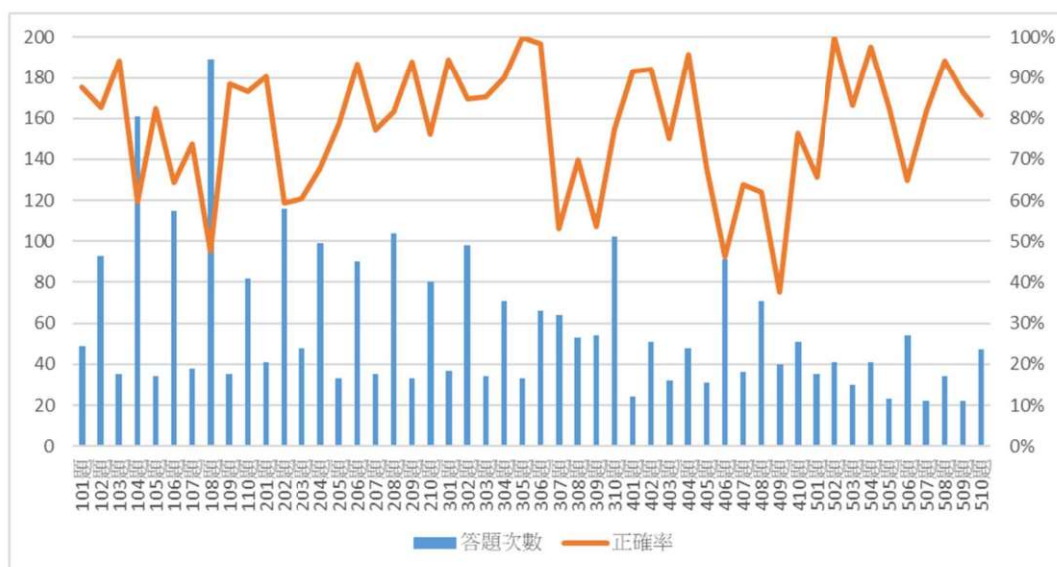


圖 7、程式自動評測平台使用統計

編號	系級	學號	姓名	108-1				108-2					
				期中考	解題數	正確率	平時成績	期末考	期中考	解題數	正確率	本學期解題數	本學期正確率
001	運動資訊	404090		100	83	92.77	22.16	100	60	183	92.35	100	92
002	運動資訊	405090		98	91	84.62	21.53	100					
003	運動資訊	405090		93	33	78.79	15.58	65					
004	運動資訊	405090		100	44	81.82	17.02	100					
005	運動資訊	405090		85	57	49.12	12.86	60					
006	運動資訊	405090											
007	運動資訊	405090							100	67	91.04	67	91.04
008	運動資訊	406090							80	5	60	5	60
009	運動資訊	408090							0	23	43.48	23	43.48
010	運動資訊	408090		60	74	64.86	16.87	100	40	241	58.09	167	55.09
011	運動資訊	408090		100	111	75.68	21.81	60	80	244	81.15	133	69.17
012	運動資訊	408090		48	82	78.05	19.69	100	80	249	64.26	167	57.49
013	運動資訊	408090		68	50	58	13.69	30	40	70	64.29	20	80
014	運動資訊	408090		98	56	71.43	16.38	100	40	129	63.57	73	57.53
015	運動資訊	408090		38	93	46.24	15.49	100	40	129	48.84	36	55.56
016	運動資訊	408090		30	69	79.71	18.84	80	20	158	71.52	89	65.17
017	運動資訊	408090		23	36	83.33	16.58	70	0	98	78.57	62	75.81
018	運動資訊	408090		45	58	34.48	10.58	85	60	248	45.16	190	48.42
019	運動資訊	408090		95	73	52.05	14.71	70	40	156	49.36	83	46.99
020	運動資訊	408090		100	56	75	16.95	100	60	184	72.28	128	71.09
021	運動資訊	408090		100	67	79.1	18.57	100	100	166	80.12	99	80.81
022	運動資訊	408090		88	51	70.59	15.81	100	100	81	72.84	30	76.67
023	運動資訊	408090		100	51	70.59	15.81	100	100	112	78.57	61	85.25
024	運動資訊	408090		71	68	60.29	15.61	50	60	128	59.38	60	58.33
025	運動資訊	408090		100	61	83.61	18.78	100					
026	運動資訊	408090		0	0	0	0.00	0					
027	運動資訊	408090		100	87	39.08	13.82	80	40	98	38.78	11	38.38
028	運動資訊	408090		0	116	56.03	19.06	30					

圖 8、平台使用與期中考、期末考成績之比較

二、學生學習成就評量

第一個學期之期中考，主要是以教科書為主的內容進行上機考試，而期末考則是以 TQC 之題庫 101~201 之奇數題為範圍，進行上機考。考試的結果如圖 9 所示，期中考有 8 位滿分，期末考共有 12 位滿分，整體而言因此階段教學範圍小且簡單，學生學習成就評量成果普遍為佳。

而第二個學期之期中考，是以 TQC 之檢定考試範圍(101~510)，採用認證考試的考場環境進行評測。考試結果如圖 10，共 20 人應考，通過(>70 分)7 人(35%)，而未通過 13 人(65%)。對比於上一學期之學習成就相對為差，其原因由學生深度訪談之回饋得知，多數學生反映，與上學期相比考試範圍太大以致成績較差。也有些學生提到，面對認證考試上機實測環境太緊張，以致失常。

1	A	B	C	D	E	F	G	H	I	J	K	L	M
2	編號	系級	學號	姓名	期中考	解題數	正確率	平時成績	期末考	學期成績			
3	001	運動資訊	404090		100	83	92.77	22.16	100	97			
4	002	運動資訊	405090		98	91	84.62	21.53	100	96			
5	003	運動資訊	405090		93	33	78.79	15.58	65	80			
6	004	運動資訊	405090		100	44	81.82	17.02	100	92			
7	005	運動資訊	405090		85	57	49.12	12.86	60	74			
8	006	運動資訊	408090		60	74	64.86	16.87	100	82			
9	007	運動資訊	408090		100	111	75.68	21.81	60	87			
10	008	運動資訊	408090		48	82	78.05	19.69	100	82			
11	009	運動資訊	408090		68	50	58	13.69	30	63			
12	010	運動資訊	408090		98	56	71.43	16.38	100	91			
13	011	運動資訊	408090		38	93	46.24	15.49	100	75			
14	012	運動資訊	408090		30	69	79.71	18.84	80	71			
15	013	運動資訊	408090		23	36	83.33	16.58	70	65			
16	014	運動資訊	408090		45	58	34.48	10.58	85	68			
17	015	運動資訊	408090		95	73	52.05	14.71	70	81			
18	016	運動資訊	408090		100	56	75	16.95	100	92			
19	017	運動資訊	408090		100	67	79.1	18.57	100	94			
20	018	運動資訊	408090		88	51	70.59	15.81	100	88			
21	019	運動資訊	408090		100	51	70.59	15.81	100	91			
22	020	運動資訊	408090		71	68	60.29	15.61	50	71			
23	021	運動資訊	408090		100	61	83.61	18.78	100	94			
24	022	運動資訊	408090		0	0	0	0.00	0				
25	023	運動資訊	408090		100	87	39.08	13.82	80	84			
26	024	運動資訊	408090		0	116	56.03	19.06	30	52			
27										81			

圖 9、第一學期之學生學習成就表現

班級名稱	學號	考生姓名	學科成績	術科成績	總分	合格狀況	出席狀況
運傳一	4080901		0	40	40	不合格	已出席
運傳一	4080901		0	80	80	合格	已出席
其他	4040905		0	60	60	不合格	已出席
運傳一	4080903		0	40	40	不合格	已出席
運傳一	4080901		0	40	40	不合格	已出席
運傳一	4080904		0	40	40	不合格	已出席
運傳一	4080903		0	100	100	合格	已出席
運傳一	4080901		0	80	80	合格	已出席
運傳一	4080901		0	40	40	不合格	已出席
運傳三	4060902		0	0	0	不合格	已出席
運傳一	4080901		0	40	40	不合格	已出席
運傳一	4080903		0	60	60	不合格	已出席
運傳四	4050905		0	80	80	合格	已出席
運傳一	4080903		0	60	60	不合格	已出席
運傳四	4050905		0	100	100	合格	已出席
運傳一	4080903		0	100	100	合格	已出席
運傳一	4080901		0	20	20	不合格	已出席
運傳一	4080901		0	0	0	不合格	已出席
運傳一	4080902		0	60	60	不合格	已出席
運傳一	4080903		0	100	100	合格	已出席

圖 10、第二學期之學生學習成就表現

三、學生問卷回饋

研究中，我們分別在第一個學期末和第二個學期末，以問卷調查的方式來蒐集學生的學習意見回饋，問卷的內容分別有：學習動機、教材教法、學習程效；以及針對本研究所採用之「程式評測平台」和「流程圖」、兩人一組之合作學習策略、學習成效。而部分的問題是以反向題方式呈現。因問卷的樣本數不足 30 分，屬小樣本因此不進行信度、效度檢定，僅統計平均分數與標準差，結果如表 2、表 3，並做為研究中深度訪談時的參考。

從表一中可看到，題號 A3 的平均分數極高，顯示出學生大都認同學習程式設計的重要，但 A1、A2 也說明學生興趣與主動性屬普通。而題號 B1~B4 的分數，反映出學生對於採用線上評測系統有極正面的回饋。而 C1、C6 可看出，學生對於課程的評量表現不慎滿意，而學習興趣未更提升。

表 2、第一學期間卷之內容與結果統計

	問題	平均	標準差
A. 學習動機			
1	我對於學習「程式設計」很有興趣。	3.75	0.97
2	我是自己主動要選修「程式設計」的課程。	3.85	1.18
3	我覺得學習「程式設計」能建立專業，對自己的將來很有幫助。	4.45	0.67
4	未來我有興趣想從事資訊相關工作	3.30	1.08
5	我選修「程式設計」課程，只是為了畢業學分，興趣其實不高。	2.50	1.19
B. 教材教法			
1	「教科書」(期中考前)與「線上評測系統」(期中考後)，我比較喜歡用「線上評測系統」來上課。	4.30	0.66
2	我覺得「線上評測系統」對於學習成效很有幫助。	4.45	0.60
3	我覺得「線上評測系統」可以激勵我，一題接著一題練習，獲得答對時的成就感。	4.60	0.50
4	我很喜歡使用「線上評測系統」這類強迫練習的上課方式，可督促自己認真上課。	4.30	0.66
5	我覺得「線上評測系統」應該是課後回家練習就好，上課應多講一些「教科書」的內容。	2.90	0.91
6	老師上課內容教太多、太快，不易吸收。	2.35	0.67
7	上課講的內容，我有很多聽不懂。	2.85	1.04

8	課堂上練習的時間不夠。	2.65	0.99
9	上課所遇到的程式問題，都有問老師、同學或靠自己研究，最後都有解決。	4.20	0.83
C. 學習成效			
1	我「期中考」得到的分數符合我應有的程度，覺得很滿意。	3.65	1.46
2	我「期末考」得到的分數符合我應有的程度，覺得很滿意。	4.15	0.81
3	修了一學期的課後，我覺得程式設計對我來說很困難。	2.90	1.41
4	我都很認真的在上課。	4.0	0.92
5	我下課後有再花時間做練習，準備考試。	4.1	0.91
6	上了一學期的程式設計課程後，我對於程式設計其他相關課程的興趣是「更提高」的。	3.65	0.88

在表 3 中，我們看到學生對於題號 A1~A5 和 B1~B5 的平均分數均超過 4 分，可顯示出採用 ADRI 的教學模式，更加深化講解程式的流程與語法，並採合作學習策略是非常被學生接受與認同的。但在 C1~C5 以及 D1~D6，則一般分數都在 4 分以下，顯示學生對於「程式設計」仍普遍覺得自己學的不夠好。但在 D2、D4、D5 上則有超過 4 分，說明學生在第二學期的課程上，已能掌握學習的方法與重點，並有認為程式設計能力有相當多的進步。

表 3、第二學期間卷之內容與結果統計

	問題	平均	標準差
A. 使用「程式評測平台」結合「流程圖」的講解：			
1	能讓我節省學習的時間	4.29	0.67
2	更快了解學習內容	4.53	0.62
3	改善我的學習效能	4.29	0.77
4	改善學習的品質	4.41	0.51
5	使的學習程式語言變得容易	4.35	0.79
B. 兩個人一組的學習方式：			
1	能讓我節省學習的時間	4.12	0.86
2	更快了解學習內容	4.06	0.90
3	改善我的學習效能	4.06	0.90
4	改善學習的品質	4.06	0.83
5	使的學習程式語言變得容易	4.24	0.83
C. 本學期學習程式設計			
1	很少有不會寫的問題	2.47	1.00
2	很少出現程式的 bug	2.00	0.87
3	很方便隨時可以自學寫程式	3.47	0.94
4	更清楚知道要如何學習寫程式	3.82	0.95
5	教學的活動(如:課後練習、期末專題)，都很容易上手	3.35	1.17
D. 學習成效			
1	我「期中考」得到的分數符合我應有的程度，覺得很滿意	3.35	1.46
2	我「期末考」得到的分數符合我應有的程度，覺得很滿意	4.00	0.79
3	修了兩個學期的課後，我覺得程式設計對我來說還是很困難	3.47	0.94
4	與上學期相比，我更認真、更投入在課程上	4.00	0.87
5	與上學期相比，我自認為學習成果有「更好」	4.12	0.70
6	與上學期相比，我對於程式設計的課程的興趣是「更提高」的	3.94	0.66

6. 建議與省思

本研究在實施兩學期課程後，提出之建議如下：

- 一、成對編程，不容易於課堂上實施。(如同組學生均不擅長程式時。)
- 二、學生人數少之課程，計畫成效難以用量化驗證，而以質性方式呈現。
- 三、各系(本校其他運動相關科系)學生，須具備的程式設計能力不同。
- 四、不同學生、不同時間的學生狀態都不同，處在一個變動環境中，成效不容易驗證。

而對於未來教學上，提出之省思如下：

- 一、藉由計畫之執行，可讓教師放下腳步，思考教材教法，仔細了解學生的學習情形。
- 二、計畫拉近了老師與學生的距離，教學評量分數有提升。
- 三、學生學習成就易受外部因素影響。
- 四、學習成就(認證考試)佳的學生，是否就具備利用撰寫程式解決問題(如：專題製作之呈現)。
- 五、本計畫之教學模式，是否適合擴展應用到全校所有科系(如全校性的通識課程)。

參考文獻

- 紹陽林, 少桓董, 宗德林, & 筱倩曾. (2017). 一個支援適性化學習的程式設計練習系統. *人文社會學報*, 13(2).
- Brito, M. A., & De Sá-Soares, F. (2014). Assessment frequency in introductory computer programming disciplines. *Computers in Human Behavior*, 30, 623-628.
- Johnson, C. N. (2016). The benefits of PDCA. *Quality Progress*, 49(1), 45.
- Malik, S. I., & Coldwell-Neilson, J. (2017a). Impact of a new teaching and learning approach in an introductory programming course. *Journal of Educational Computing Research*, 55(6), 789-819.
- Malik, S. I., & Coldwell-Neilson, J. (2017b). A model for teaching an introductory programming course using ADRI. *Education and Information Technologies*, 22(3), 1089-1120.
- Malik, S. I. (2018). Improvements in introductory programming course: Action research insights and outcomes. *Systemic Practice and Action Research*, 1-20.
- Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2018). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education*, (99), 1-14.
- Palominos, F. E., Palominos, S. K., Durán, C. A., Córdova, F. M., & Díaz, H. (2018). Challenges in the use of a support tool with automated review in student learning of programming courses. *Procedia computer science*, 139, 424-431.
- Guo, X. Y., Wen, S. T., & Liu, Y. G. (2017). Towards Improving the Practical Ability by Teaching Mode Reform of Courses of Programming. *DEStech Transactions on Social Science, Education and Human Science*.
- Vihavainen, A., Airaksinen, J., & Watson, C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. *Proceedings of the tenth annual conference on International computing education research*, 19-26.
- Razvi, S., Trevor-Roper, S., Goodliffe, T., Al-Habsi, F., & Al-Rawahi, A. (2012). Evolution of OAAA strategic planning: using ADRI as an analytical tool to review its activities and strategic planning. *Proceedings of Seventh Annual International Conference on Strategic Planning for Quality Assurance and Accreditation of Universities and Educational Arab Institutions*.

附件一：學生深度訪談逐字稿

姓名 (編號)	問題	回答
L01	你上學期成績都考一百分，那這一次期中考就只有考六十，最主要原因可能是什麼？	考試太緊張，加上可能考試準備範圍比較大一點，然後題目準備的方式比較沒有像以前那樣子比較集中，因為以前只要一到三章，這次一到五章，然後多了整整二十題要做。
	緊張部分，之前考試也是有用評測平臺考試，為什麼這次還是更緊張？	這次可能因為有牽扯到證照的部分，得失心會更重一點。
	還是說，時間到了會有倒數計時是這些影響嗎？就評測平臺的環境有沒有影響？	多少有啦，多少都是有一點，就看到那個時間越來越少，心情就會多少會受到一些影響。
	練習的次數跟上課的態度應該沒什麼變？練習一百次、正確率九十二	有些是我就在那邊亂套，我想說，就是有點像是想要鑽漏洞的感覺，就想說，我可以用多麼簡單的方式寫這個系統有辦法接受。
	正確率跟上學期差不多啦，沒有變	會錯就是我在鑽漏洞啦，我可以透過簡單的方式寫，系統有辦法接受這樣。
H01	我這邊看你成績，你上學期期末可以考到一百，那你這學期評測平臺解題數算滿多的，但是準確度不是很高，你覺得你考這個成績，主要是因為為甚麼？	因為我一、二、三單元都還 ok，沒有看得很仔細，一、二是有穩答，後面三、四、五就是後來緊張啊，然後就忘記了。
	所以你是考場作答到後面兩題時間感覺又快用完會緊張是這樣嗎？	對，差不多，三、四、五這三題我就想不出來，就想說，好，那就先放著，先寫下一題這樣。
	上學期你可以考到一百分，那時候就不會緊張，原因是差在哪邊？	因為那個時候範圍很小，就很清楚，這樣做做做。
	所以大範圍來講，對你是有壓力的，但是你在學習上，其實上學期跟這學期沒有差很多，只是因為考試範圍變大？	然後四、五章也比較難懂。
	比較難懂，你覺得你四、五算有懂嗎？還是本來題目就很難？	五比較難懂一點，四 OK。
	那上課練習當中你覺得那個五很難嗎？	嗯。
	很難，好不容易做出來，但是考試還是就會記不住嗎？	還是會，就是會某幾項忘記了。
K01	那你上學期期中考考九十八期末考考一百，這學期你考試時覺得是差在哪裡？	熟悉度跟範圍，熟悉度的話就是因為我是為了過才去碰的，平常就不會碰，因為我算是死背，就是沒有特別去深入，熟悉那些程式碼，這次考試範圍又是整個合在一起，自己準備時間也沒有說抓得很好。
	所以你上學期的成績還不錯，分數還蠻高，主要是你剛剛提到的，你主要是用比較偏背誦、記憶的方式去了解程式，實際上，整個邏輯要讓你自然而然記起來比較不容易，是你個人對科目的差異是不是？這樣講對嗎？	算是。
	所以範圍太大就會變得不容易理解，對不起來？	因為資訊會很多。
	資訊很多，那你覺得你這部分有辦法去克服嗎？還是以前對於數學這種科目就會比較不擅長？	以前都是補習補出來的，能夠去瞭解...應該說都是當下能夠了解，但是過一段時間可能就是會忘記一些。
所以程式也是這樣子是不是，就是當下可能有懂，但是記不起來？	就是需要比較長的時間去了解。	

H01	你上學期期中考得不是很理想，但是期末分數就有拉回來，啊你這學期這次考試又考得不是那麼好了，我看你練習的次數好像也是降低了，這樣看起來是努力程度不夠是這樣嗎？還是有什麼原因？	期初跟這學期這時候打工比較忙，所以就比較少練習。
	打工比較忙，但是看來你應該是有潛力，是可以考很好的，你對自己會有信心嗎？	蠻有信心。
	蠻有信心，只是你花的時間不是那麼多，是這樣嗎？看你有沒有給自己設定一個類似門檻啦，比如說你再怎麼忙，還是要抽出時間要拼到比如說六十分、五十分，還是等考試完再看看是這樣嗎？你的想法？	想說應該，感覺憑印象應該可以考得還不錯，結果就破功。
	那你上學期期末怎麼拼可以拚到一百分？	就常常練，上學期期末我練習很多次。
	花了很多心思，那你這次跟上學期期末比時間比起來大概是二比一還是差很多？	差很多。
	F01	我們探討一下，你考二十現在算是倒數第三，是什麼原因呢？
	所以是操作介面，覺得讓你手忙腳亂不熟悉？	沒有，我原本已經做答對兩題了，然後最後我自己把第一題答案，就是我又再送一個答案上去，然後就蓋掉了我第一次的答案，所以我也嚇到。
	所以至少應該要四十分就對了啦？	嗯，我有問那個人說可不可以幫我從後臺找出我原本的答案，可是他說不行。
	不行啦，因為蓋掉就蓋掉了，那你自己覺得你如果是考 40，這樣算滿意嗎？	不滿意，至少要有 60。
	你覺得你花的時間，這樣子比起來算多還是少？	我自己覺得我花很多時間，但是就是考這個的時候，還是會緊張又...，前面幾題都還可以，但是後面就...，就忘記了。
	那上課的練習，覺得有聽懂嗎？還是其實上課就聽不太懂？	上課也聽不懂，有一些啦。
	大部分都懂，還是一點點不懂？	有幾題比較...，像那個函式跟迴圈有時候邏輯還是，就會忘記說接下來要怎麼辦。
	我們大概教二十題有了嗎？每類十題，假設函式這樣有十題你大概有幾題不太懂？	有一些簡單的，但是可能三、四題吧。
	十題裡面有三四題比較不懂，那如果考出那三四題就是會就不會對就對了？那如果考到其他七題就會對了？	對。
	所以比例大概是三題？	而且我自己是覺得...
	考卷可能有一份是比較簡單的？	現在說也沒有用啊，就是這樣。
T01	你考試考起來四十分，這個分數覺得滿意嗎？	很不滿意，因為我花很多時間練，可是後來又忘記了。
	那你是覺得回家花很多時間還是課堂上花很多時間？都有？	都有，因為我自己空堂的時候也會到圖書館練，然後就是把那些五顆星、三顆星那種星星比較高的都多練幾次這樣。
	我看你第一次考試成績九十五上學期，還算滿好的期末是七十是不錯的，但是現在感覺好像越來越往下降，是因為這樣從上學期到現在覺得越來越難了，還是什麼原因？還是越聽越不懂？還是越教	越聽越不懂吧，就是變的很複雜。

	越難，還是?	
	所以前面的內容你覺得比較簡單?	比較好理解。
	後面教到函式跟迴圈是這個比較聽不懂是嗎?	函式比較難。
	比較難的章節還是聽不懂?	就是我就...不太理解。
	無法理解就對了不曉得為什麼會這樣為什麼要這樣寫啦?這樣是嗎?	嗯。
	那有加那個流程圖有沒有幫助?還是其實流程圖也看不懂?	看不太懂，因為他都是一直分分支出去，所以我覺得寫得有點太複雜。
	所以有看流程圖跟沒看其實差異不大?因為流程圖本身就很難懂?這樣講對嗎?	就是他一開始主幹然後旁邊分支出來定義函數那些，然後再接回去主幹，但是因為程式碼寫起來是先定義函數，然後再...，感覺順序...。
	順序還不太能接受就對了啦?	無法理解。
	那你以前高中數學好不好?	中上。
	中上，所以太難的數學，以前學這種數學太難的會不會有挫折感?	都會有，但是都有解決啊。
	那像這個程式後面最近考試這個算有解決嗎?還是只有少數幾題還是不會而已，還是大部分考出來都不會?還是有那種差一點點，其實就是差一點點而已?	都是差幾個步驟，但是出來的結果就是差蠻大的，就是大部分都記得，但是可能就漏，漏寫什麼。
	就少了一、兩、一兩小步?	對。
	然後讓你整題沒得分?	就當下想不出來。
	每題都起頭都會寫，但是到後面就是差一行或兩行沒寫對而已?是這個狀況?	差不多，不然就是前面可能有漏寫一、兩個步驟，然後後面寫的是對的，然後就執行不出來這樣。
K02	那你上學期第一次考一百，第二次期末考八十，現在這一次期中考考四十，這樣看來好像本來表現很好，越來越往下降，是什麼原因呢?	基本的可能比較熟悉一點，再加上因為上學期就是本身是沒有在打工，現在是有在打工，所以六日會比較忙一點，所以練習的時間就有壓縮到，這次考試就是想說至少把很基本的把握住，就是不要放掉，可能有些人會覺得說三四五不太會，那可能就是連同之前的可能跟著放棄，那我是說，至少我希望我能把握住基本的就好。
	所以你後來對的、作答對的是一、二類是這樣嗎?假設一二三四五就是把握?	可能一、二、三，就是想說一、二、三比較有把握。
	四、五因為時間不太夠，就有點放棄啦?	對，可是還是有去看一下，可能就是沒辦法到像一、二、三這麼的熟悉。
	那上課，上課時間，因為你說你下課比較沒時間準備，那上課的練習對你來講，會不會變得就非常重要?但是這邊看起來你上課，好像練習也不多，還是其實你就是用那一種簡單的、基本分的策略，因為上課要休息，是這樣嗎?	也不能這樣說，就是因為我覺得後面聽起來，真的是就是有點聽不太懂，所以就說，那就至少把握住自己會的就好。
	所以你四、五類其實是聽不太懂的?	對。
	那以前上學期的一、二類是聽完全懂嗎?	也不能說是完全懂，就是略懂，至少會比起四、五讀起來比較輕鬆。
	應該說讀了保證就有分?	對。
	四、五類是讀了之後可能還是看不懂?	對阿。
	啊就沒有分，是這個意思嗎?	嗯。
	就不值得投資就對了啦?	對。

	那四、五類後來加了流程圖，這個對你有沒有什麼幫助?還是流程圖也是看不懂?	就是有些還是看不太懂。
	加流程圖還是不容易理解，對不對?	對。
L01	那你這邊解題數，這邊看起來，你是練習的次數蠻多的?	就惡補這樣。
	你有很努力的練習，考出六十分，你覺得滿不滿意?	我覺得還可以，因為另外兩題自己沒有把握好，然後練習，就難度比較高的自己沒有把握好，自己沒有多練一點。
	是四、五類那邊的題目嗎?	對。
	一樣是比較後面比較覺得比較有難度啦?	對。
	那你覺得上課四、五類，你覺得有困難的部分是懂的程度到哪裡?似懂非懂?還是其實上課時候有看懂哪一題，但是第二次再拿出來考會不記得?是到什麼程度?	第四題其實自己應該要會，就是有做大概有做出來，可是就是沒有做對這樣。
	所以可能有一行寫錯了，一直找不出問題，但是整個邏輯是懂的?	對。
	那第五類的錯的題目的話?	第五類就是自己比較不會這樣。
	上課就不會了?	上課我有聽懂，但是我就回去做，就是做不太出來。
	就重新自己寫是寫不出來的，但是看老師寫的程式碼是都聽得懂?	對。
	那個流程圖是看得懂嗎?	第五題的第四跟第五，第四題是 ok 的，第五題就有點不瞭解。
	就平常上課的流程圖，第五類還是看不懂就對了?	對。
	這樣應該表現還不錯啦，因為看你的練習次數是最多的，回家花多少時間練習?考前惡補嗎?	前一個禮拜惡補這樣，然後就比較早起然後練到晚上。
	一整天喔?	就中午稍微休息，然後就練一整天，大概是這樣。
L02	你上學期都考試成績都考一百分都滿分，啊你這學期怎麼突然就掉下來了?	準備範圍。
	範圍變大?	對。
	那你的跟上學期自己比起來的努力程度有差嗎?準備時間?	應該是有，因為這學期就是有活動蠻多的，就是時間搭配上可能有點不太均勻這樣。
	但是我看你練習次數還是算在前面，不是太低，還是你是課堂上很努力在聽?	我就是回宿舍練習比較多。
	回宿舍也是有練，那考不好的，也不是不好啦，就是差一點點那個原因是怎麼樣?那兩題錯的是會寫但是差一兩行還是完全無法知道從哪裡開始寫?	應該是第一題跟第四題，第一題的話就是我寫出來然後程式跑出來對，然後我上傳之後那個批改程式就是寫錯誤這樣子，第四題的話可能就是自己寫到應該是剩最後一行，然後就是時間到來不及交這樣子。
	第一題你覺得應該是對的?	就是 spyder 跑出來是沒問題的，也符合題目的要求這樣子，可是批改就是錯誤這樣子。
	所以可能是之前練習說，有時候多一個空白或少一個空白那種看不太出來的小問題啦?這樣講對啦?	對。
	所以如果是這樣可能原先考到 80 那就只差一題一小行而已?	對。
	所以落差就有點不小心就變差 40 分?	對。
	那第四類第五類你上課都聽得懂嗎?	後面因為有加那個流程圖就是比較容易能夠了解，大概那個程式要怎樣子寫這樣子。

	所以流程圖對你是覺得還 ok?	還不錯。
C01	你考試 60 分這樣你覺得滿意嗎?你前幾次考試好像也差不多都是這個成績啦。	不是很滿意，因為我第四題就是，我四五類比較不熟，然後我第四題就是已經快要寫出來了，但是我把 1 寫成 i，所以就算不出來，我在那邊一直想，然後還是想不出來。
	那 1 為甚麼寫成 i，是沒打好還是?	不是，因為我想說，他不是代那個 range 下去算嗎，然後我就把就想說就果我把他算出來之後，每個 i 都變成 1 的話，那就把他加回去那是不是就會，一個加，就是加 1 加 1 加 1，結果他就是只有加那個 1。
	所以你的邏輯是有想通但是寫出來的不如你預期啦?	對。
	那你還有錯哪一邊呢?	第五題不會寫。
	那你上課這樣第五類對你來講是比較困難是不是?	嗯。
	那上課的當下有聽懂嗎?	有的有聽懂。
	那大概第五類有十題你不懂的有幾題?	不懂得大概兩三題。
	啊考試剛好就是考那不懂的?	嗯。
	啊不懂是完全就不知道從何寫嗎?因為有的人是差一兩行就錯了，啊你是第五類是剛好那幾題是完全不懂就連一行都不太寫得出來嗎?	就是上課的時候就是可以理解，可是可能再寫的時候就會忘記。
	就從 0 開始就寫不出來啦，但是有提示就看得懂啦?	對。
	那那個流程圖對第四類第五類的難題有幫助嗎?	我覺得多少是有幫助，就是他比較比較讓人，就是讓我覺得可以就是他的關係，順下來的順序會比較順。
	知道起承轉合從哪邊開始寫就對了?	對。

附件二：專家諮詢會議會議記錄

時間、地點、與會者
<p>開會時間：中華民國 109 年 6 月 11 日（星期四）下午 14 時 0 分</p> <p>開會地點：採 Csico Webex 線上會議</p> <p>出席專家學者：</p> <p>林晉榮 教授（國立中正大學運動競技學系暨運動與休閒教育碩士班）</p> <p>姜林杰佑 教授（國立高雄科技大學金融資訊系）</p> <p>主持人：許育嘉 博士</p> <p>會議記錄：顏晨聿</p>
專家意見整理
<p>林教授意見整理：</p> <ol style="list-style-type: none">1. 資訊是未來的趨勢，建議將程式設計列為兩個學期的必修課程，讓學生更清楚課程內容，將課程列為必修也會讓學生對自己的要求更高。2. 第一學期利用應用軟體，如:office、TQC 作課程的引導並加入資料庫，配合課堂作業，減少學生對於程式設計的恐懼感，第二學期做比較應用的實作課程，如架設網站，讓體育運動的學生了解基本的概念，知道需要時可以使用甚麼工具完成作業。3. 在課程名稱上進行調整如:運動與資訊處理、運動與資料分析，提升學生興趣。4. 問卷的反向題的問題需要再更精準，分數看起來並不高，整體的分數看起來偏低，從學習動機來看或許一開始就有一些壓力，可在課程安排做一些調整。5. 實作比理論更實用，建議用實作的方式讓學生間接學習，傳統教學、理論的東西學生的接受度較不高。6. 上課做課堂作業，邊教邊做，課堂練習在課堂結束前需繳交，這樣可以間接的觀察學生的學習程度，再加上回家作業，透過回家作業了解學生遇到的問題，再於課堂中與學生討論，提升學生的對於課程的吸收度。7. 課堂人數越少，學習效果應該會越好，如果人數過多可能吸收會沒那麼好。8. 課堂人數少，建議主要利用質性研究，問卷只是參考。9. 體育運動科系的學生主要著重在應用，建議可參考 TQC，TQC 有很多類別，可訂定目標讓學生達到一定程度，是否要考取證照可視學生學習程度而定。 <p>姜林教授意見整理：</p> <ol style="list-style-type: none">1. 建議將基礎程式設計列為必修，進階程式設計列為選修，才不會造成程度參差不齊、進度落後跟不上的問題，可錄製教學影片或是開設加強課程，讓沒有基礎的同學進行先修，將程度補齊後再進入課程，練習的題目建議分為基礎題及應用題。2. 學生是一個遊戲世代，建議設計遊戲，利用破關的方式讓學生具備基礎程式的能力，較為困難的題目可利用問題導向建立學習動機，從所學領域激發學生動機，例如透過資料分析帶入課程，再加入程式設計，學習更深入更有效率的分析。3. 第一學期建立基礎能力，第二學期建議分組，去訂定與所學領域相關之主題，並透過分析資料達成學習之目的。4. 成對編程前提是需要兩個人能力是對稱的，一個人負責編碼、一個人負責檢查，但學生之間較難找到對稱的組員，建議可分 3~4 個人一組製作小組專題，組員內可能有人對於編成較為熟悉，可帶領程度較差的學生一起互相學習。5. 每一堂課設計一個題目，解完的學生可提早下課，但題目難易度不能太難，透過立即

的動機提升學生的學習動機，需要配合助教檢查學生是否有確實完成題目。

6. 不同領域的學生需要的資訊能力並不一樣，運動相關科系的學生較不需要開發前端的能力，反而資料處理的能力較為重要。
7. 學生程度落差太大的問題建議在課程開始前先進行分流。
8. 鼓勵參賽，有時候參賽能提升學生的學習動機，讓學生共同完成一個專題，但參賽可能無法要求全班都參加，如果是通識課程的話更為困難，因為每個系所要求的另不同，需要再探討相關問題。