

中華大學生物資訊學系系統開發專題報告
高中資訊科技科銜接教育課程教材規劃與製作
Planning and production of textbooks for senior high school IT courses

專題組員:李典衛、簡士豪、連建勳、溫瑞泓

專題編號:105005

指導老師:侯玉松老師

1. 摘要

教育部自 108 學年度起實施新課綱,108~110 學年度入學的高一新生約 30 萬人需接受 16 小時的資訊銜接課程。但是教育部官方教材存在程式語言不一致等缺點,本專題以教育部官方教材為基礎設計出二套 6 小時實體課程教材,包含榜單查詢與幾何圖形繪圖程式實作等,程式語言分別為 Python 與 Blockly,改進教育部官方教材的缺點,並在竹南高中實施,透過前後測問卷分析,顯示使用本專題製作教材確有教學成效。

關鍵字:程式設計、銜接課程、教材規劃

2. 簡介

在資訊科技廣泛的全球下資訊科技日漸被重視,學習程式語言將像英文一樣成為現代人的必修。

依據教育部 108 年學年度高級中等學校新舊課綱銜接教育實施計畫裡,要求 108~110 學年度入學的高一新生適用新課綱,在舊課綱、新課綱教材中會有所差別,因此需要新課綱銜接課程,預計有 30 萬人必須修讀 [1]。

在新課綱銜接課程中,資訊銜接課程共 16 小時,包括 Python 等程式語言 2 小時、演算法 5 小時及程

式設計 9 小時,其中程式設計 6 小時要實體上課 [1],因竹南高級中學一屆有 14 個班級,全校只有兩間電腦教室,7 天要上完銜接課程,學校的師資不足,為此請求中華大學來幫忙教 6 小時的實體課程

生物資訊系侯玉松老師與林志陽老師及資工系其他老師指派到竹南高中協助教學,發現教育部提供的教材還有很大的改進空間,缺點如下

(1)教育部教材使用的程式語言不一致。

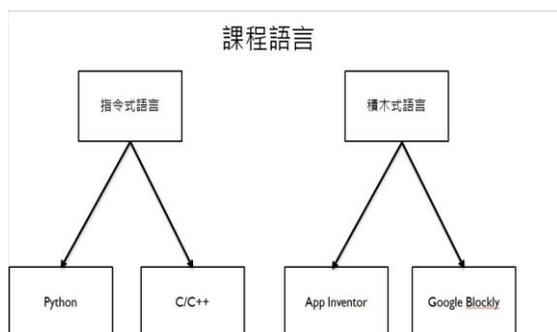
(2)教育部假設學生已看完前 10 小時數位課程教學且部分章節只有附程式碼,沒有解釋。

本專題著手研究克服上述缺點,改良教育部資訊銜接課程教材,希望能提供未來 109、110 學年度入學之高一新生使用。

3. 專題進行方式

(1)教育部官方教材指定程式語言:

教育部官方教材指定使用四套語言為:Google Blockly、App Inventor、Python 和 C/C++做為高中銜接課程之授課語言 [2],可分為指令式語言與積木式語言二大類,指令式語言包括 Python、C/C++,積木式語言又包括 Google Blockly、App Inventor(如圖一)。



圖一、銜接課程程式語言分類圖

(2)選擇程式語言發展教材:

近年來 Python 語言為 IEEE Spectrum 票選為第一名[3]，也因為竹南高中選擇使用 Python 語言為主做為教學，選擇 Google Blockly 語言，則是因為撰寫程式容易並且可以產生 JavaScript, Python, PHP 或 Dart 等原代碼。

(3)教材設計:

A. 教育部教材規範:

教育部的教材規範，語言(數位課程 2 節)、演算法(數位課程 5 節)、程式設計(3 節數位與 6 節實體課程)，10 節數位課程的部分由學生自行上網觀看學習，6 節實體課程必須到學校電腦教室，由老師教導學生程式設計實作。

語言部分介紹四種語言、Google Blockly、App Inventor、Python 和 C/C++。

演算法部分介紹基本概念、陣列資料結構及演算法介紹。

程式設計部分介紹程式語言基本概念、功能及應用(數位課程 1 節)、結構化程式設計(數位課程 2 節+實體課程 1 節)、陣列程式設計(實體課程 2 節)、模組化程式設計概念及問題解決(實體課程 3 節)。

B. 教育部規範的實體課程內容

(A)結構化程式設計

a、循序結構

b、選擇結構:包含單一、雙重、多重選擇結構

c、重複結構:包含計數式與條件式迴圈

(B)陣列程式設計實作

(C)模組化程式設計概念、模組化程式設計與問題解決實作

(4)教材內容規劃

在實務上，高一新生極少在上課前事先觀看教育部的數位課程，在實體課程教材裡，必須將演算法的基本概念、陣列結構的演算法概念與應用及 Python 語言的基本概念、功能濃縮在教材的第一部分與第二部分裡。由於只有六個小時的實體課程，需要濃縮二個小時介紹數位課程的基本概念，使得學生能夠銜接到實體課程，故將教材分為五個部分如下。

第一部分:介紹演算法、運算思維和程式設計。

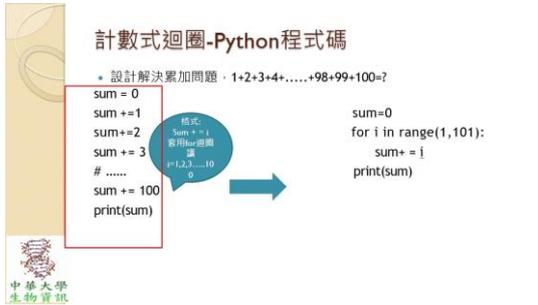
第二部分:介紹 Python 與 Google Blockly 程式開發環境與語言的基本概念，包括:變數、資料型態、運算式、輸出指令。

在程式開發環境方面，Python IDLE 的安裝程式較小，Anaconda 較大，為了節省安裝時間，採用 IDLE 做為程式開發環境。Google Blockly 程式開發環境則使用[4]。

第三部分:教育部的教材結構化程式部分，分為兩節數位課程與一節實體課程，內容中包含循序、選擇、重複三種基本結構，實作案例採用相同於教育部的 BMI 程式設計。

重覆結構分為計數式迴(for)、條件式迴圈(while)，此部分與教育部不同的部分在於，強調計數式迴圈定義

為確定要重複做某個特定數則採用計數式迴圈(以計算 $1+2+\dots+100$ 為實作案例, 如圖二), 條件式迴圈為不確定要執行幾次則採用條件式迴圈。(以猜數字為實作案例, 如圖三)



圖二、計算式迴圈範例程式碼



圖三、條件式迴圈範例程式碼

第四部分: 陣列程式設計的部分, 教育部教材的實作案例為榜單查詢程式設計, 但是只有展示 Blockly 的程式設計, 忽略了 Python 程式語言的部分, 為彌補程式解析不足, 我們完成 Python 對應程式碼。(如圖四)



圖四、陣列程式設計範例程式碼

第五部分: 模組化程式設計概念及問題解決, 我們採用幾何圖形練習讓同學們在學習將大問題逐步化成有規律的小方法解決這個大問題(如圖五)。



圖五、模組化問題解決

(5)主要困難與解決之道:

A、在教材設計上, 需要把 16 小時的課程濃縮在 6 小時的課程教材內是有難度的, 且部分學生線上數位課程未觀看, 可能導致教學內容無法理解。

解決方式:

以實體課程的程式案例著眼, 將所需的先備知識從 10 小時的數位課程中抽離濃縮為 2 小時, 內容請參閱第 3 之(4)節的第一、二部分。

再將 6 小時實體課程濃縮為 4 小時, 內容請參閱第 3 之(4)節的第三、四、五部分。

B、教學設備硬體方面倘若使用教育部教材, 安裝檔案過於龐大, 同時安裝會使時間過長。

解決方式:

Python 採用 IDLE 平台以縮短原先程式載入時間, 如果安裝 Anaconda 亦可使用。

C、對於本專題生而言, 大一至大四的課程內容中, 大多以 C++ 做為主要教學內容, 不熟悉銜接教材中以 Python、Blockly 為主要語言。

解決方式:

不熟悉兩種語言的我們，需花時間去學習 Python 和 Blockly 語言，一邊學習，一邊製作教材。

D、教育部提供之教材上，重點教學過於複雜，導致程式碼出現還沒教過的指令(例如:if 指令的範例程式出現 def 等指令)。

解決方式:

教學教材內修訂程式碼，避免程式碼出現還沒教過的指令。



圖六、循序結構

4. 主要成果

專題與教育部銜接教材結合做出一套 Python 教材及 Blockly 教材。

第一、二部分:為彌補學生線上 10 小時課程當中有模糊的地方在此做為複習，採用部分教育部教材之 PPT 與本專題生自行製作之 PPT 做為整合此部分中的概念介紹。

第三部分:結構化程式設計，覺得教育部教材這部分有許多不宜之處，在介紹循序、選擇結構時，教育部教材上的程式碼用了還未學到的函式與後面才會教到的模組化程式設計(如圖六)，在這本專題運用此部分所要學的概念做解釋，而不會運用到還未學到的程式碼做為教學範例。在重複結構這部分，包含計算式迴圈與條件式迴圈，此部分教育部教材內的範例為 $1+2+\dots+100$ ，本專題採用與其相同的範例，不同的是教育部教材內只有程式碼並未解釋其原理，而本專題生在範例上有做其解釋與原理為何，在條件式迴圈這部分教育部採用 $1+2+\dots+N$ 為範例，本專題生認為一種是確定重複次數的而另一種是不能確定重複次數的，進而採用猜數字做為這部分的範例。

第四部份:陣列程式設計實作

上，教育部範例只有 Blockly 之語言解釋，在專題上採用 Python 做為解釋語言，並附上程式構想與說明。(如圖七)



圖七、陣列結構

第五部分:模組化程式設計與問題解決實作上，教育部教材採用正多邊形的實作範例，對應教材寫出程式碼，且衍生較需思考的問題。(如圖八)



圖八、模組化程式設計

上述教材內容統整出比較表(如表一)

表一、教材比較表

程式名稱	教育雲教材	Python	Blockly	備註
Scratch	有	0	0	
Scratch內嵌	0	0	0	教育雲教材內有Scratch，Scratch內有Scratch
Python	有	0	0	Scratch在Python中特別編譯Scratch
Scratch	0	0	0	Scratch在Python中特別編譯Scratch
Scratch	0	0	0	Scratch在Python中特別編譯Scratch
Scratch	有	0	0	Scratch在Python中特別編譯Scratch
Scratch	0	0	0	Scratch在Python中特別編譯Scratch
Scratch	0	0	0	Scratch在Python中特別編譯Scratch
Scratch	0	0	0	Scratch在Python中特別編譯Scratch

問卷測驗：

108年8月20日在竹南高中實施教學，為得知教學成效製作了一份前後測問卷(如圖九)，回收問卷數為32，有效問卷為32，無效問卷為0。

問卷中設計13個選擇題，每題設有五個選項為：非常同意、同意、無意見、不同意、非常不同意，分數分別設定為5分、4分、3分、2分、1分，以做為觀察學生是否有進步。

高中資訊銜接課程學習成效問卷調查

您好！
我們是中華大學的學生，首先感謝您填寫本問卷調查，這次的問卷調查分成前後測兩種不記名方式，主要是想瞭解同學們對於程式設計的一些認知與想法，請同學們依照個人主觀的想法來填寫，請留意每題都要完成，敬請放心作答。再次感謝您的幫忙與支持！

敬祝：
學業進步 身體健康

中華大學生物資訊學系
指導老師 侯玉松教授

前測(後測)

1. 知道程式設計在做什麼
2. 以前有學習過程式設計課程
3. 理解演算法的意義
4. 理解演算思維的意義
5. 知道程式設計「變數」的意義
6. 知道輸入指令與輸出指令的功能
7. 會將一元二次方程式的根公式寫成程式
8. 知道單一選擇、雙重選擇、多重選擇的差異
9. 瞭解程式流程圖的意義
10. 知道計數式迴圈與條件式迴圈的差異
11. 知道程式設計「陣列」的意義
12. 知道程式設計「函式」的意義
13. 知道程式設計「函式帶參數」與「不帶參數」的差異
14. 簡潔的描述自己對於程式設計的認知是什麼(至少10個字)

非常
非
常
同
意
同
意
無
意
見
不
同
意
非
常
不
同
意

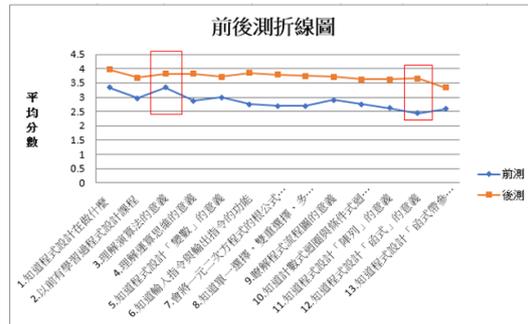
本次的測驗到此結束 再次感謝您的配合

圖九、問卷設計圖

從前後測問卷分數中得知，普遍學習都有進步，代表教學上有顯著的提升，而起伏最大的是知道程式設計「函式」的意義，代表學生在這部分學習進步較多，而在理解演算法這部分，學生學習上可能較為模糊不懂，與其他部份相比差距較低，這也讓我們知道往後教材要在該部分在做修改。(如表二、圖十、十一)

表二、問卷測驗項目

項目	前測平均	後測平均
1.知道程式設計在做什麼	3.34378	3.96875
2.以前有學習過程式設計課程	2.96875	3.6875
3.理解演算法的意義	3.34375	3.8125
4.理解演算思維的意義	2.875	3.8125
5.知道程式設計「變數」的意義	3	3.71875
6.知道輸入指令與輸出指令的功能	2.75	3.84375
7.會將一元二次方程式的根公式寫成程式	2.6875	3.78125
8.知道單一選擇、雙重選擇、多重選擇的差異	2.6875	3.75
9.瞭解程式流程圖的意義	2.90625	3.71875
10.知道計數式迴圈與條件式迴圈的差異	2.75	3.625
11.知道程式設計「陣列」的意義	2.625	3.625
12.知道程式設計「函式」的意義	2.4375	3.65625
13.知道程式設計「函式帶參數」與「不帶參數」的差異	2.59375	3.34375
總平均	2.84375231	3.71875



圖十、問卷測驗結果

學生回饋：

學生 A:「利用程式設計，將複雜的問題一一分解，進而有效率的解決或完成問題及反覆的運算。」

學生 B:「對於程式有了一定程度的認識，如果有定式和參考就能夠做出一些成果。」

實際教學情形:(如圖十一)



圖十一、教學照片

5. 評估與展望

本專題經試教的過程後，從問卷中評估發現演算法部分對於高中生而言可能較難吸收，亦或是教材內容設計上較為難理解，因此在這部分需要很大的改進空間，之後或許能採用更多圖像來介紹演算法概念。

目前實施對象和時間的不足，此專題在優化教材中還有很大的進步空間且不成熟，期望在優化上更加成熟，讓後續使用的教師能快速且方便，也讓學生了解教材內容。

6. 結語

在製作專題中，有許多困難之處，例如：教材設計、撰寫報告內容、影片拍攝、海報製作，因為都是首次接觸，而其中出現各種問題，包括(製作專題段落、圖文關係、問卷設計等)，若參考此專題演算法部分時，應注意教材選用之重點整理，對於學生理解方面需要多加留意授課內容。

7. 銘謝

感謝侯玉松老師對此專題的教

導，讓我們參與高中銜接教材的製作，在途中有遇到困難，老師都會與我們討論，不厭其煩的解答，老師常有事但還是配合我們的時間與我們討論，謝謝老師犧牲自己的時間與我們一起製作了一份銜接高中的教材。

8. 參考文獻

[1]聯合新聞網，史無前例！8月下旬起3年30萬高一新生要上新課網銜接課程

(<https://udn.com/news/story/7266/3941824>)

[2]教育部國民及學前教育署新課網銜接教材數位平臺下載專區

(https://ecc.pro.edu.tw/TR_Material/index#1)

[3] iThome，IEEE Spectrum 發布程式語言排名，Python 持續強勢、R 語言逐漸消退

<https://www.ithome.com.tw/news/125008>

[4]Blockly

<https://simulator.webduino.io/>