

新北市113年度國中小資訊科技優良教案徵選實施計畫

教案設計

服務學校	新北市立板橋國民中學		設計者	黃文耀
參加組別	<input checked="" type="checkbox"/> 程式教育組 <input type="checkbox"/> 人工智慧組		<input type="checkbox"/> 資訊素養與倫理組	
領域/科目	科技領域/資訊科技		實施年級	7
單元名稱	流程控制結構 Flow Control Structure		總節數	共 8 節， 360 分鐘
設計依據				
學習重點	學習表現	<ul style="list-style-type: none"> ● 運 t-IV-4能應用運算思維解析問題。 ● 運 p-IV-1能選用適當的資訊科技組織思維，並進行有效的表達。 ● 設 c-IV-3能具備與人溝通、協調、合作的能力。 	核心素養	<ul style="list-style-type: none"> ● 科-J-A2運用科技工具，理解與歸納問題，進而提出簡易的解決之道。 ● 科-J-B1具備運用科技符號與運算思維進行日常生活的表達與溝通。
	學習內容	<ul style="list-style-type: none"> ● 資 A-IV-1演算法基本概念。 ● 資 P-IV-1程式語言基本概念、功能及應用。 		
議題融入	實質內涵	<ul style="list-style-type: none"> ● 性 J3檢視家庭、學校、職場中基於性別刻板印象產生的偏見與歧視。 		
	所融入之學習重點	<ul style="list-style-type: none"> ● 探討學校教育及職場中，資訊科技領域，男女表現、從業人數、成就等差異。 		
與其他領域/科目的連結	<ul style="list-style-type: none"> ● 數學領域 			
教材來源	<ul style="list-style-type: none"> ● 康軒資訊科技1上課本[3] ● 翰林資訊科技1上課本[4] ● 南一資訊科技1上課本[5] 			
教學設備/資源	<ul style="list-style-type: none"> ● 電腦 ● 投影機 			
使用軟體、數位資源或APP內容	<ul style="list-style-type: none"> ● Scratch 線上版 ● MS PPT ● Google Classroom 			
學習目標				
<ol style="list-style-type: none"> 1. 能理解資訊科技是學習「以運算思維解決問題」的一門科目。 2. 能理解流程控制的概念：以結構化的流程控制結構表達演算法，可以增加可讀性、容易修改及方便除錯。 3. 能夠掌握三種流程控制結構： <ul style="list-style-type: none"> ● 循序結構(Sequence Structure); ● 選擇結構(Selection Structure); ● 重複結構(Repetition Structure)。 4. 能利用流程控制結構完成指定任務：分析任務=>以流程圖表達演算法=>以 Scratch 撰寫程式=>執行程式讓電腦完成任務。 				

5. 能繪製符合流程圖繪製原則的流程圖：
- 標準符號(Flowchart Symbols)¹；
 - 簡潔明確；
 - 繪製方向由上而下，由左至右；
 - 流向線避免交叉或過長。
6. 能依照流程圖，使用相對應的 Scratch 積木及程式結構，完成結構化程式。

¹參考7上資訊課本(康軒「流程圖圖示說明」及翰林「流程圖的符號與功能說明」)。

教學活動設計

教學活動內容及實施方式

時間

使用軟體、數位資源或 APP 內容

第1節 運算思維與演算法

1 運算思維(Computational Thinking)

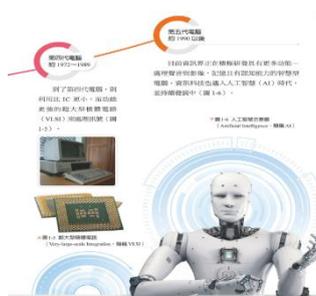
- 請同學閱讀康軒1上資訊科技課本第2頁，找出資訊科技所學何事？ --- 每一冊資訊科技課本第2頁第1句話，**資訊科技是學習「以運算思維解決問題」的一門科目。**
- 說明同學運算思維就是「**電腦科學家面對問題的思維模式**」，及其重要性。
 - 以康軒1上資訊科技課本讓同學體會30年來資訊科技帶來的生活變革。
 - 以康軒1上生活科技課本簡述工業革命、工業4.0帶入物聯網及電腦智慧化。
 - 以翰林1上資訊科技課本讓同學了解目前已進入人工智慧的時代，面對未來的環境，須具備「**電腦科學家面對問題的思維模式**」，也就是「**運算思維**」。

25

1. 電子書
 - 翰林1上資訊科技課本
 - 康軒1上資訊科技課本
 - 南一1上資訊科技課本
2. 翰林1上資訊科技範例程式
3. Scratch 線上平台
4. Google classroom
5. Microsoft Powerpoint
6. Chrome
7. YouTube

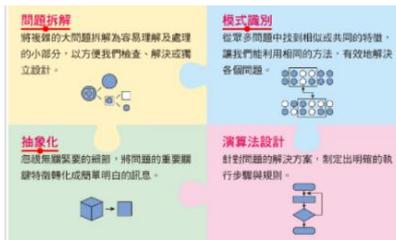


項目	1784	1870	1969	Now
發展條件	第1次工業革命	第2次工業革命	第3次工業革命	工業4.0
影響	蒸氣機	電力	電腦	物聯網
	機械化	生產線	自動化	智慧化



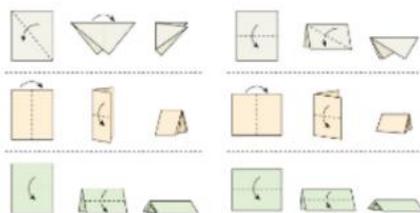
- 導讀康軒1上資訊科技課本第2章的科技廣角，介紹第一位提出「**運算思維**」的華裔美籍電腦科學家 --- 周以真博士，探討資訊產業界男女比例差距議題，勉勵同學(尤其是女同學)，有為者亦若是。

- 簡介運算思維並說明運算思維的四個基石(Conerstone)。
 - (1) Decomposition(問題拆解)
 - (2) Pattern Recognition(模式識別)
 - (3) Abstraction(抽象化)
 - (4) Algorithm Design(演算法設計)



2 演算法(Algorithm)

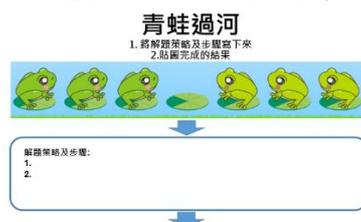
- 說明演算法是解決問題的方法。
- 舉例日常生活中的演算法(e.g., 正確洗手5步驟、自動販賣機)^[5], 說明演算法就是解決問題的方法。
- 讓同學體驗「對折再折」活動^[4], 說明精確指令才能確保正確的成果。



對
的
結
果

- 觀看並導讀 BBC Learning WHAT IS AN ALGORITHM^[6]。
- 電腦科學家所設計的演算法就是電腦解決問題的方法，必須更為嚴謹才能讓電腦順利且正確的執行，進而歸納出演算法的特性^[5]：

- 分組玩「青蛙過河」遊戲^[7], 寫出完成任務的步驟(演算法), 確保能執行, 且必然產生結果, 繳交作業¹。



¹ 作業範本請參照：附錄1 作業：演算法_青蛙過河.pptx
學生作品請參照：圖1-1~圖1-4。

第2節 流程圖 (Flow Chart)

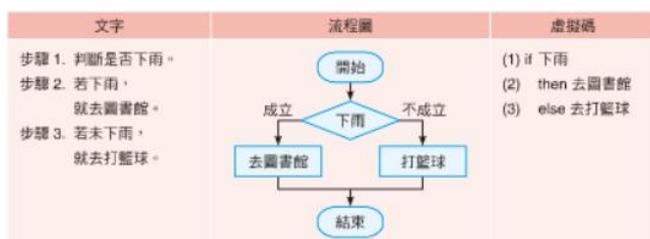
1. 在資訊科技領域，程式語言是讓電腦實踐演算法的工具，而給電腦的演算法必需是清楚而符合邏輯的步驟。

20

2. 簡介演算法的表示方式，並舉例說明流程圖有可讀性高

3. 容易理解的優點^[4]。

- 文字
- 流程圖
- 虛擬碼



4. 簡介流程圖符號^[3]

符 號	意 義	說 明
	開始 / 結束	流程圖開始或結束
	處理	處理一項工作
	流程方向	流程進行的方向
	輸入 / 輸出	進行資料輸入或輸出
	決策	依條件比較結果進行不同處理
	迴圈	迴圈變數初值與終值的描述
	連接	其內書寫編號，作為流程的連結點

5. 流程圖繪製實作

- 教學：以 Microsoft PowerPoint 工具列之「常用-繪圖 - 圖案 - 流程圖」功能，繪製流程圖符號。
- 實作：學生能找到圖形及線條，編輯文字並更改格式。

20

6. 說明流程圖繪製原則^[4]

流程圖繪製原則

1. 使用標準符號，以便閱讀和分析。
2. 文字簡潔且明確可行。
3. 繪製方向由上而下，由左至右。
4. 流程線條避免交叉或過長。



5

備註：在此先快速簡介流程圖符號及繪製原則，後續實作繪製流程圖，將不斷加強

<p>複習。</p> <h3>第3節 流程控制結構(Flow Control Structure)</h3> <p>1. 打造演算法三種主要結構控制結構^[4]。</p> <ul style="list-style-type: none"> ● 循序結構(Sequence Structure) ● 選擇結構(Selection Structure) ● 重複結構(Repetition Structure)  <p>2. Code.org/特快課程 2021版本¹</p> <p>特快課程 2021版本共28階段，為一個完整的程式設計學程，導入主要目的是成為自主學習平台：</p> <ul style="list-style-type: none"> ➢ 登入並修改個人資料。 ➢ 概述28階段，對應上述3個流程控制結構。 ➢ 對於程度好的同學，當課堂指定作業完成後，可自主學習，超前學習，亦可補足課堂沒教的程式設計技巧。 ➢ 對於程度中等的同學，可作為複習與加強程式撰寫技巧的工具。 ➢ 對於程度較差或學習有障礙的同學，可按部就班緩慢前進，或自己挑選感興趣的關卡，完成任務。 ➢ 這個課程會成為同學的寒暑假作業，希望每天撥出一點時間，接觸程式，訓練運算思維。 <p>特快課程 2021 階段1: 與憤怒鳥一起編程</p> <p>學生使用憤怒鳥遊戲中的角色開發循序演算法，讓憤怒鳥能夠從迷宮的一側移動到另一側的豬那裡。要完成這項操作，學生就需要順序疊起程式碼積木，讓憤怒鳥向前、向左或是向右移動。</p> <p>這一個階段總共有13關，於正式教授演算法與程式設計前導入，主要目的：</p> <ul style="list-style-type: none"> ➢ 階段1可作為一個前測，了解同學程度。 ➢ 利用階段1第13關，解釋演算法的差異，及設計演算法之思維模式。 <p>3. 運算思維的最後一哩路：好的演算法</p> <p>同學於小學時，或多或少接觸過積木程式，主要學習的方式偏重於引起興趣與問題解決，鮮少接受按部就班，有系統的教學，寫出來的程式只求能動，能解決當下問題(最好是有趣的)，進入國中後，希望的是能讓學生有結構化的思維模式，將眼光放長遠，建立好的演算法/程式重於對的演算法/程式的態度。</p>	<p>10</p> <p>10</p> <p>15</p> <p>10</p>	
--	---	--

上述階段1第13關是一個很好的範例，當班上有3~5位同學完成第13關時，安排觀摩同學程式，並講解何謂好的程式。



「這隻豬真的把我惹毛了！」
憤怒鳥如何用9塊積木捉到這隻豬呢？

1	<p>蒙著頭做的同學，第一個遇到的問題是，就是發現竟然要13塊積木才能完成</p>	<p>當運行時</p>
2	<p>程度好的同學，可以用迴圈使用9塊積木完成任務(2-1)，稱讚之餘，可鼓勵同學想想看，是否可以再少些積木。 這時有可能出現8塊積木完成任務(2-2)。</p>	<div style="display: flex; justify-content: space-around;"> <div data-bbox="742 869 877 1120"> <p>2-1</p> <p>當運行時</p> </div> <div data-bbox="877 869 1088 1120"> <p>2-2</p> <p>當運行時</p> </div> </div>
3	<p>有些同學無能力使用迴圈，但看到提示了，用9塊積木完成任務了。</p>	<p>當運行時</p>

檢討這幾個程式的重點是：哪個程式較好?用什麼評斷方式?

- A. 空間：哪個程式用的積木少?
- B. 時間：哪個程式更快捉到小豬?
- C. 複雜：哪個程式簡單?

基本上有爭議的是2-2和3:

- ✓ 2-2往前走，憤怒鳥移動步數較多，程式也較複雜，但積木較少(8塊)，
- ✓ 3則多一塊積木(9塊)，但程式較直覺、簡單，憤怒鳥移動步數較少，可較快捉到小豬。

這裡必須教導的觀念是：上述 A、B、C 是好的演算法必須考量的基本項目，有好的演算法才會有好的程式。至於2-2和3，正如同要從台北到高雄，坐高鐵或坐台鐵好呢?必非絕對，必須取捨。

那有又快又好的程式嗎?

演示以5塊積木完成任務²，用憤怒鳥的股去撞小豬，同學大笑之餘，印象深刻。

總結，希望同學思考每一個任務的解方案時，將上述 A、B、C 時刻放腦中，覆推敲，精益求精，寫出好的程式，不只是完成任務的低標而已。



屁
決
反
而

¹於 code.org 開立班級，同步 google classroom，同學以新北市政府 google 帳號登入。

²少數同學程度好，做得快，可以提示，是否挑戰用5塊積木完成任務，還是有同學做得出來，演示時將更有說服力。

第4節 循序結構(Sequence Structure)

1 觀念解說

- 說明循序結構的流程圖架構，並舉例說明¹
- 說明一個敘述代表一個程式碼，以 Scratch 而言，可想像是一行積木。
- 程式由上而下依序執行，就稱為循序結構。

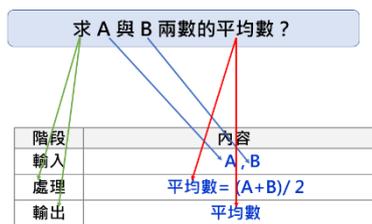


10

2 完成翰林課本2-3-1範例：求 A 與 B 兩數的平均數。

(1)問題分析：

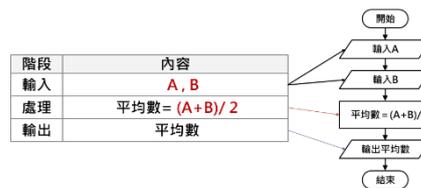
- 以課本表格為鷹架，著重在任務中找出關鍵字，放入表格中。
- 精簡文字
- 將數學的表達方式改為程式人的表達方式
- 課本抄寫參考圖2-1
- PPT 作業參考圖2-2



10

(2)流程圖：

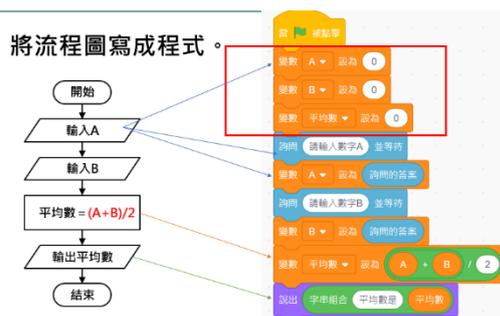
- 修改課本流程圖文字，著重在正確使用流程圖符號，以流程圖忠實表達並實踐問題分析結果。
- 課本抄寫參考圖2-1
- PPT 作業參考圖2-2



10

(3)程式：

- 課本程式加入 **變數初始化**，建立同學良好的程式設計模型(建立變數=>初始化=>詢問=>將詢問的答案指定給變數)，以這個範例而言，建議設為 0，重複執行時亦可明確觀察變數變化²。



10

- 教導相關 Scratch 積木。
- 要求程式要執行驗證。
- 作業參考圖2-2

總結此一階段須達成的目標：

- (1) 讓同學開始導正之前一股腦寫程式的習慣，改為 top down 的思維模式，由「理解任務說明=>分析問題=>以流程圖表達演算法=>程式設計」，這必須重第一支程式做起，期待1年後能養成這個習慣。
- (2) 了解每一階段的關聯性與一致性。
- (3) 熟悉流程圖符號
- (4) 熟悉以 PPT 繪製流程圖
- (5) 熟悉 Scratch 積木

5

¹本章節主要以翰林版課本^[3]為主，其優點是針對每一個控制結構有組織有層次的分別介紹，每一控制結構完整地包含問題分析、流程圖到程式實作，可以建立同學完整而正確地的思維模式。

²翰林1上課本並不重視變數的初始化，時有時無，甚至設定值錯誤，建議同學一開始寫程式就養成變數初始化的好習慣，而且要知道如何設定對的初始值，

此處可參考康軒版教科書^[4]。

第5節 循序結構(Sequence Structure)

1 檢討同學作業

2 作業訂正

3 發布精熟題給已完成課本範例的同學

任務：輸入段考國、數、英、社、自等5科成績，算出平均分數，然後說出平均分數。

4 說明並實作精熟題

(1)問題分析：

- 著重於由任務說明的一段文字找出關鍵字(名詞)與關鍵動作(動詞)。
- 作業參考圖2-3

(2)流程圖

- 仿作：複製上一題流程圖，利用相同的架構，修改成本次任務的流程圖。
- 導入流程圖的连接符號，為後續較複雜的流程圖鋪路。
- 作業參考圖2-3

(3)程式

- 仿作：教導複製角色的技巧，複製上一題角色，利用相同的積木與程式架構，修改成本次任務的程式。
- 要求對 Scratch 專案及角色命名，名稱對了再寫程式，養成好習慣。
- 要求程式要執行驗證。
- 作業參考圖2-4

精熟題純屬仿作，以複製及修改即可完成，實作技巧並不難，也不費時，主要在修正觀念與養成正確習慣。但務必讓同學先完成課本範例(基本上是抄襲啦!)，建立良好基礎，而不必要求精熟題一定要100%完成。

程度好的同學完成作業後，到 code.org 特快課程自主學習。

15

10

10

10

第6、7、8節 選擇結構

1 觀念解說

(1)說明選擇結構是依條件判斷結果，決定執行哪一段程式。

(2)條件判斷的結果只有：真(True)或假(False)2種¹。

(3)定義

●敘述：一個程式碼，以 Scratch 而言，可對應成一行積木。

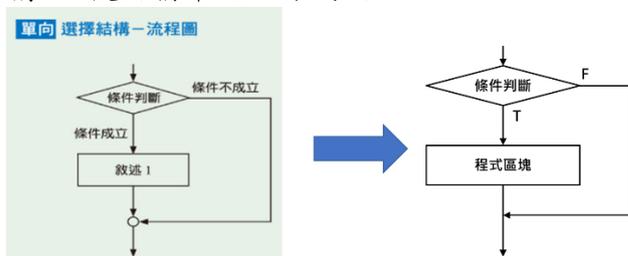
●程式區塊²：一段程式碼，亦即由多個程式敘述組成。

(4)選擇結構又可分為三種

- 單向選擇結構。
- 雙向選擇結構。
- 多向選擇結構。

2 單向選擇結構

首先導入判斷符號(菱形)，修改課本單向選擇結構之流程圖架構，使用精準而正確的用語。



課本抄寫參考圖3-1

翰林課本並無單向選擇結構範例，自行新增以「輸入資訊考試成績，判別是否及格，不及格就抄考卷」為任務，讓同學以上個單元學到的方法(問題分析=>流程圖=>程式)，加上新學到單向選擇結構的語法、流程圖符號及程式積木，完城任務。

(1) 問題分析：

- 導入單向選擇結構的語法：如果……，則……。
- 根據學到的單向選擇結構流程架構圖，完成任務。
- PPT 作業參考圖3-2

(2) 流程圖：

- PPT 作業參考圖3-3

(3) 程式：

- 教導相關 Scratch 中相對應的程式積木。
- 程式要求要輸入60、95、59.5，執行驗證。
- 作業參考圖3-4

¹翰林版課本^[3]此處使用「條件成立」、「條件不成立」來代表 True 和 False，而在3上課本會再複習選擇結構，則使用 True 和 False；康軒版1上課本^[4]則使用「成立」、「不成立」；南一版1上課本^[5]則使用「是」、「否」；於邏輯運算說明時，又使用「成立」、「不成立」、「○」、「×」，徒增困擾，故而統一使用「True」和「False」，使用於流程圖或真假值表時則簡寫成「T」、「F」，使用正確而精準的專有名詞，有利於溝通與後續銜接數學與資訊課程。

²翰林1上課本以敘述取代程式區塊，是錯誤的用詞，翰林3上課本，相同的內容則使用正確的用詞。此外康軒版1上課本^[4]並未犯此錯誤，故而修正，杜絕錯誤觀念。

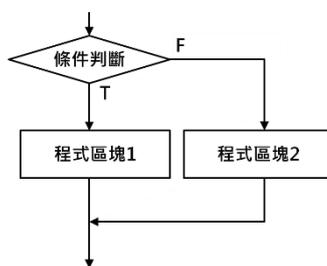
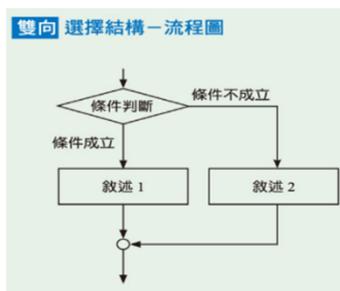
10

10

25

3 雙向選擇結構

請同學依據上一個單元修改單向選擇結構的流程圖架構，修改課本雙向選擇結構之流程圖架構。



10

完成翰林課本3-2範例：

下列為改寫後之任務敘述，讓同學更容易將問題拆解為輸入、處理、輸出、判斷、輸出五個階段，並寫出各階段相對應的內容。

35

【範例】

輸入各項成績，換算為學期成績，輸出學期成績，判斷學期成績是否及格，輸出學期成績及格或不及格？
(作業成績40%、測驗成績40%、平時表現20%、學期成績60分為及格分數。)

(1) 問題分析：

- 雙向選擇結構語法：如果……，則……，否則……。
- 根據學到的雙向選擇結構流程架構圖，完成任務。
- 課本作業參考圖3-5
- PPT 作業參考圖3-6

(2) 流程圖：

- PPT 作業參考圖3-6

(3) 程式：

- 教導相關 Scratch 中相對應的程式積木。
- 變數需要初始化。
- 程式要求要輸入不同數字，執行驗證。
- PPT 作業參考圖3-7

延伸探討

- 1 單向選擇結構與雙向選擇結構之使用時機。學生常會使用多個單向選擇結構來取代雙向選擇結構，這是不合邏輯的，請同學想一想為什麼？避免想出這種邏輯錯誤的程式。學生作業參考圖3-8。
- 2 程式語法的優化，可以寫出較好的程式，比如說「<60」可以換成「>60或=60」，甚至「<60不成立」，要採用哪一個呢？學生作業參考圖3-9。

思維模式仍離不開好程式的 ABC，這是要不斷提醒學生的：

- A. 空間：哪個程式用的積木少？
- B. 時間：哪個程式更快捉到小豬？
- C. 複雜：哪個程式簡單？

4 多向選擇結構

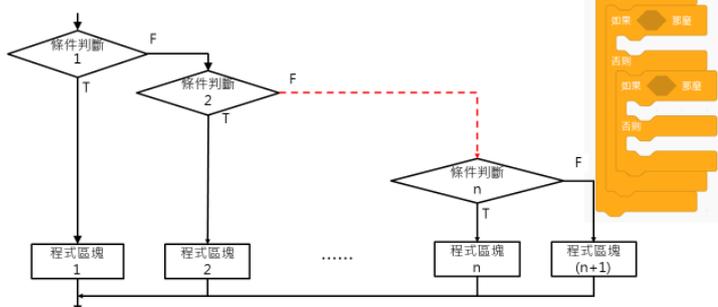
翰林課本1上並未介紹多向選擇結構，康軒1上課本則以購票及成績等第介紹多向選擇結構，故自行補充，幾例說明多向選擇結構的使用時機，讓同學完整學習，若仍不了解，則待翰林課本3上再次闡述。

10

講解多向選擇結構的流程圖及程式結構

10

多向選擇結構(巢狀選擇結構)

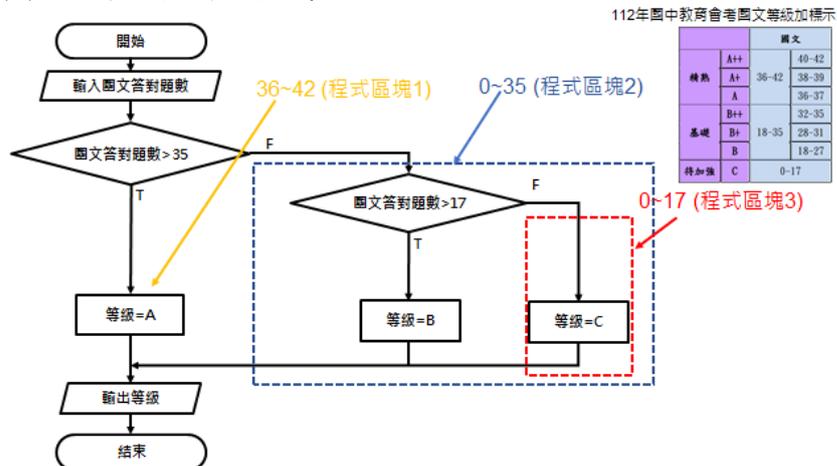


完成會考成績等級標示任務：

25

根據112年國中教育會考各科等級加標示與答對題數對照表，寫一個程式，輸入會考國文答對題數，判斷並說出 "A、B、C三等級"

- (1) 提供112年國中會考等級加標示與答對題數對照表(取自台灣師範大學心測中心官網)。
- (2) 只作流程圖及程式



- (3) 學生作業參照圖3~8、3~9、3~10、3~11、3~12

第9、10、11、12節 重複結構

1 說明何謂重複結構

10

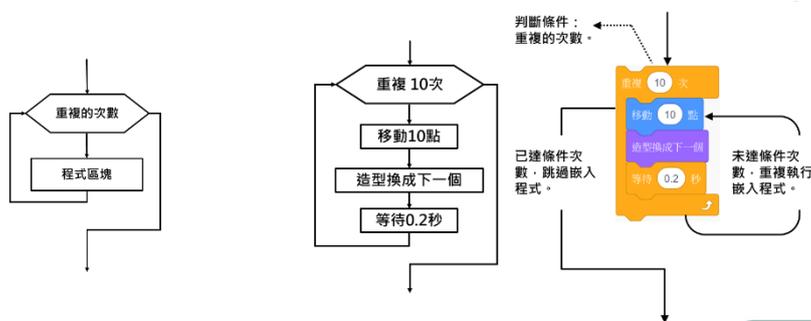
- (1) 簡介 Scratch 相關積木
- (2) 以之前學過貓狗對話中，小貓咪走路為例，導入重複執行的概念。
- (3) 將重複結構分為
 - 計次迴圈
 - 條件式迴圈



2 計次迴圈(翰林課本2-3-3^[3])

- (1) 導入迴圈之流程圖符號(六邊形)。
- (2) 修改課本計次迴圈流程圖的架構，使之流向線符合由左至右，由上而下之原則，並統一為順時針方向。
- (3) 將「敘述」改為「程式區塊」。
- (4) 完成貓咪走路的流程圖。

10



3 Scratch 程式設計 - 繪圖篇(翰林課本2-4^[3])

- (1) 簡介 Scratch 擴展功能中之畫筆相關積木。

- 教授筆跡清除、下筆與停筆三塊積木之必要性與相對位置。
- 修正翰林課本2-4-1中，「筆跡全部清除」位置錯誤，將「筆跡全部清除」放於「下筆」之上，正確架構如右圖。
- 其餘相關畫筆積木則由同學自行嘗試。

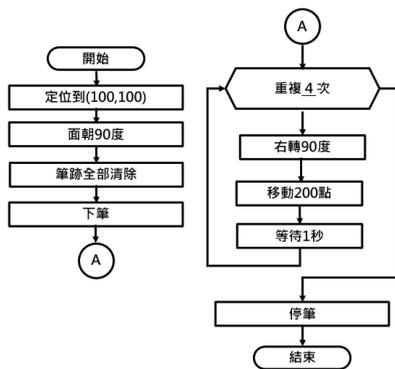


5

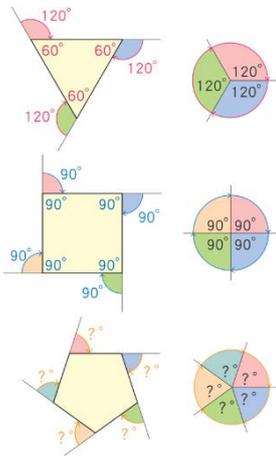
- (2) 以繪製正方形為例，說明循序結構與重複結構之差異。
- (3) 當眼光放遠，繪製邊數多的正多邊形或邊數是未知數的多邊形，則重複結構之優勢更為明顯。
- (4) 實作畫正方形

10

畫正方形



(5)教授正多邊形外角和為360度，請同學自行繪製一個正多邊形。作業參照圖4-1、圖4-2



	4	5	6	...	n
多邊形的邊數	4	5	6	...	n
三角形的個數	2	3	4	...	$n - 2$
內角和的算式	$2 \times 180^\circ$	$3 \times 180^\circ$	$4 \times 180^\circ$...	$(n - 2) \times 180^\circ$
正多邊形的一個內角	$\frac{2 \times 180^\circ}{4}$	$\frac{3 \times 180^\circ}{5}$	$\frac{4 \times 180^\circ}{6}$...	$\frac{(n - 2) \times 180^\circ}{n}$
外角和	360°	360°	360°	...	360°
正多邊形的一個外角	$\frac{360^\circ}{4}$	$\frac{360^\circ}{5}$	$\frac{360^\circ}{6}$...	$\frac{360^\circ}{n}$

15

4 Scratch 程式設計 - 計算篇 - 重複結構(翰林課本2-3-3^[3])

(1)完成範例 1+2+3+4

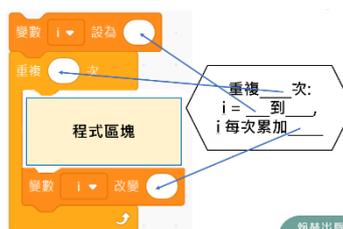
- 說明變數的概念。



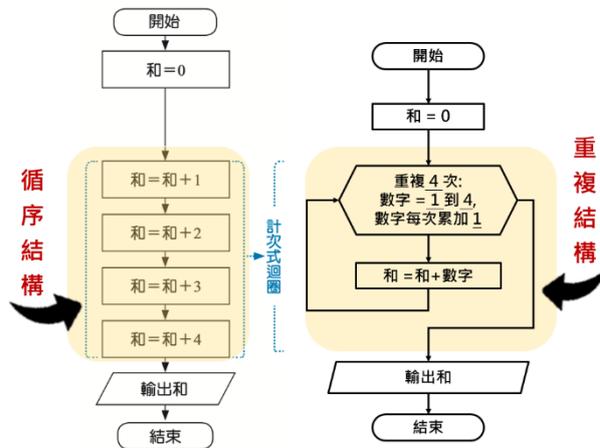
- 說明程式中"="與數學符號之差異
- 說明 Scratch 積木中改變和設為再加減法之應用。



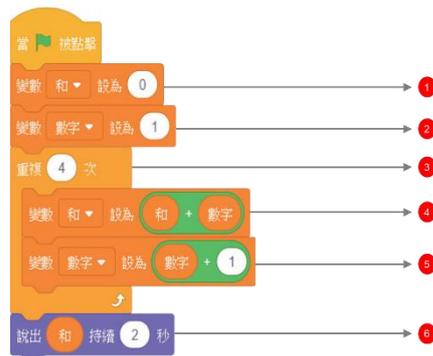
- 導入流程圖中，含有變數之迴圈符號架構與語法。
- 說明流程圖符號與程式語言之對應關係。



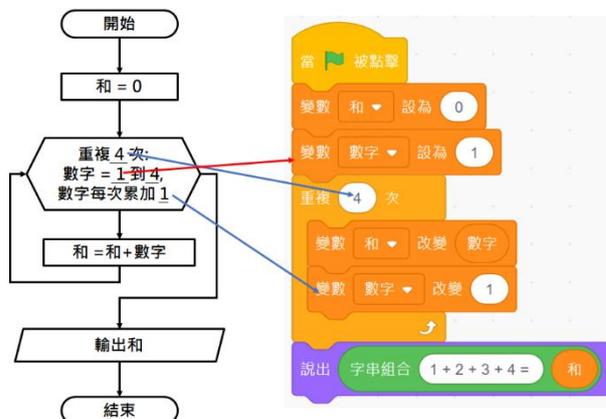
- 說明此題使用循序結構與使用重複結構之差異。
- 修改課本流程圖，使之流向線符合由左至右，由上而下之原則，並統一為順時針方向。



- 當眼光放遠，數字多或數字是未知數的情況下，則重複結構之優勢更為明顯。
- 修改課本程式，因為課本的變數「數字」初始值為0，明顯是錯的，不知道此變數之用意。



- 修改課本程式，使用「改變」代「替設」為積木，減少積木數，使程式更簡潔、省攻堅、不易錯。



(2)完成2+3+4+5

- 此範例主要用來證明課本將變數「數字」初始值設為0之謬誤。

(3)完成3+5+7+9

(4)完成1+2+.....+N

(5)完成1*2*.....*N

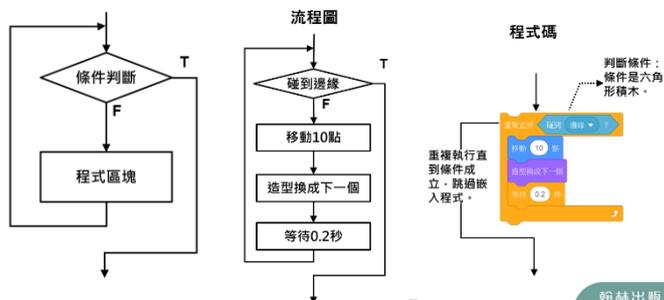
第12節 小遊戲 - 賽馬

1 說明條件式迴圈(翰林1上課本2-3)

2 修改流程圖架構

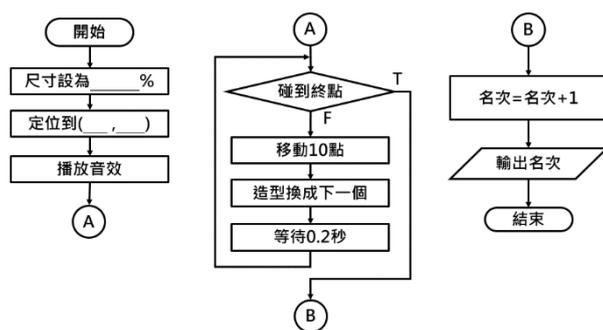
條件式迴圈

使用時機：
不知道某段敘述
要被重複執行的
次數，但知道終
止情況時，會使
用條件式迴圈。



20

3 完成翰林1下課本5-1-2賽馬遊戲



25

25

教學
成果
教學
成果

青蛙過河

貼圖完成的結果, 並將解題策略及步驟寫下來



解題策略及步驟:

1. 5到4	6. 7到6	11. 4到2
2. 3到5	7. 5到7	12. 6到4
3. 2到3	8. 3到5	13. 8到6
4. 4到2	9. 1到3	14. 3到5
5. 6到4	10. 2到1	15. 4到3



說明: 圖1-1 演算法

青蛙過河

貼圖完成的結果, 並將解題策略及步驟寫下來



解題策略及步驟:

1. 左一青蛙跳到第四格
2. 右一青蛙跳到第三格
3. 右二青蛙跳到第五格
4. 左一青蛙跳到第六格
5. 左二青蛙跳到第四格
6. 左三青蛙跳到第二格
7. 右一青蛙跳到第一格
8. 右二青蛙跳到第三格
9. 右三青蛙跳到第五格
10. 左一青蛙跳到第七格
11. 左二青蛙跳到第六格
12. 左三青蛙跳到第四格
13. 右一青蛙跳到第二格
14. 右二青蛙跳到第三格
15. 右三青蛙跳到第五格



說明: 圖1-2 演算法

青蛙過河

貼圖完成的結果, 並將解題策略及步驟寫下來



解題策略及步驟:

1. 淺綠深綠交叉
2. 淺1跳 深1跳過淺1 淺2往前跳淺1 往前跳淺2 往前跳淺3 往前跳 深123往前跳 淺123往前跳



說明: 圖1-3 演算法

青蛙過河

貼圖完成的結果, 並將解題策略及步驟寫下來

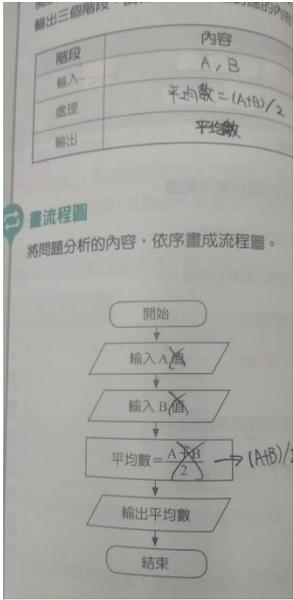


解題策略及步驟:

1. 不要同顏色再一起
2. L1-R1-R2-L1-L2-L3-R1-R2-R3-L1-L2-L3-R2-R3-L3



說明: 圖1-4 演算法



輸出三個階段

階段	內容
輸入	A, B
處理	平均數 = $(A+B)/2$
輸出	平均數

二 畫流程圖

將問題分析的內容, 依序畫成流程圖。

```

    graph TD
      Start([開始]) --> InputA[/輸入 A/]
      InputA --> InputB[/輸入 B/]
      InputB --> Process[平均數 =  $(A+B)/2$ ]
      Process --> Output[/輸出平均數/]
      Output --> End([結束])
  
```

說明: Fig. 2-1 循序結構 課本範例

平均數-問題分析

P50

階段	內容
輸入	A, B
處理	平均數 = $(A+B)/2$
輸出	平均數

平均數-流程圖

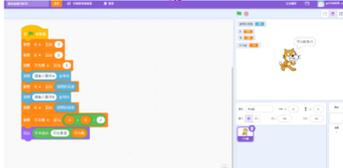
利用課本第 29 頁的符號畫出, 畫出平均數的演算法

```

    graph TD
      Start([開始]) --> InputA[/輸入 A/]
      InputA --> InputB[/輸入 B/]
      InputB --> Process[平均數 =  $(A+B)/2$ ]
      Process --> Output[/輸出平均數/]
      Output --> End([結束])
  
```

平均數-程式

P51

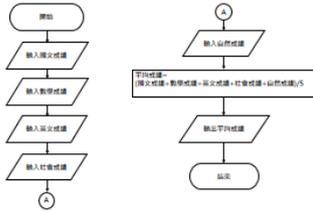


說明: Fig. 2-2 循序結構 課本範例

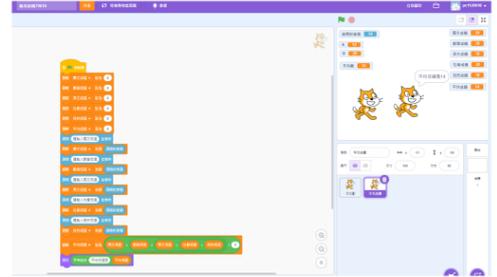
平均分數-問題分析

階段	內容
輸入	國文成績,數學成績,英文成績,社會成績,自然成績
處理	平均分數 = (國文成績+數學成績+英文成績+社會成績+自然成績)/5
輸出	平均分數

平均分數-流程圖

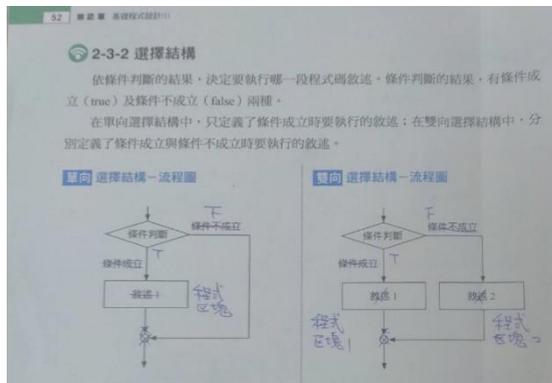


平均分數-程式



說明: Fig. 2-3 循序結構 精熟題

說明: Fig. 2-4 循序結構 精熟題



分析

階段	內容
輸入	資訊考試成績
判斷	如果資訊考試成績 < 60, 則不及格
輸出	抄考卷

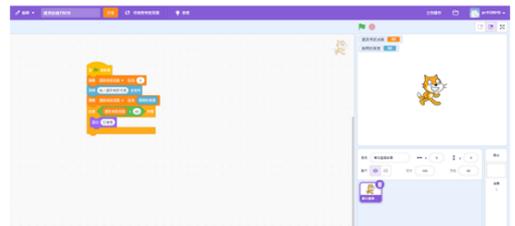
說明: Fig. 3-1 選擇結構 流程圖架構

說明: Fig. 3-2 單向選擇結構

流程圖



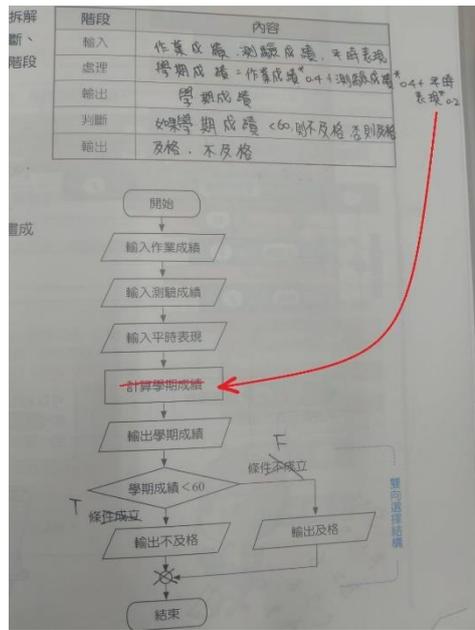
程式 (輸入60)



作業繳交3張圖，輸入60、95、59.5

說明: Fig. 3-3 單向選擇結構

說明: Fig. 3-4 單向選擇結構



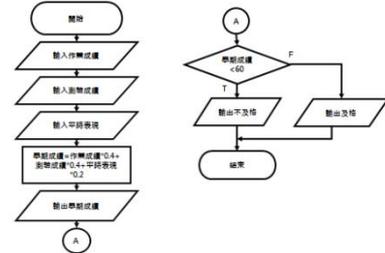
問題分析

P33

階段	內容
輸入	作業成績, 測驗成績, 平時表現
處理	學期成績 = 作業成績 * 0.4 + 測驗成績 * 0.4 + 平時表現 * 0.2
輸出	學期成績
判斷	如果學期成績 < 60, 則不及格, 否則及格
輸出	及格, 不及格

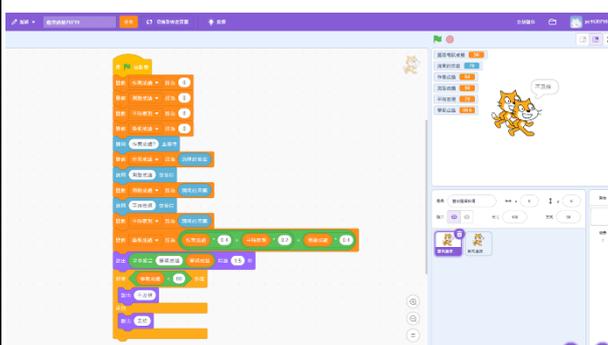
流程圖

P33



說明: Fig. 3-5 雙向選擇結構

說明: Fig. 3-6 雙向選擇結構



想一想

學期成績 > 60 或 學期成績 < 60 哪個

1. 左右兩個程式執行的結果會相同嗎?

2. 我會想用哪一個程式呢? 為什麼?

我的答案

1. 會

2. 左邊那個。左邊較簡單, 也不容易出錯, 使用的積木較少, 執行比較快。右邊程式不符合邏輯, 比如一個人考了90分, 卻還得判斷是否等於60分, 再判斷是否小於60分。

說明: Fig. 3-7 雙向選擇結構

說明: Fig. 3-8 雙向選擇結構

想一想

學期成績 > 60 或 學期成績 < 60 哪個

1. 左右兩個程式執行的結果會相同嗎?

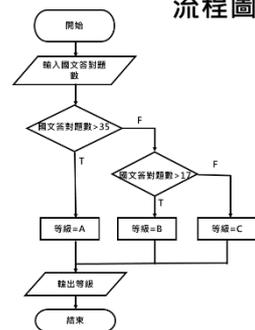
2. 我會想用哪一個程式呢? 為什麼?

我的答案

1. 會

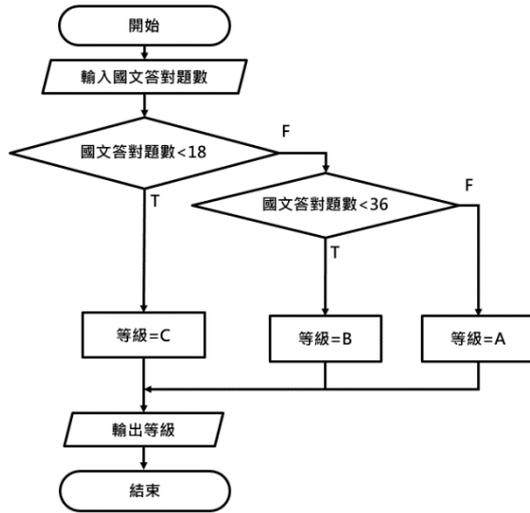
2. 左邊那個。因為較簡單, 省空間, 不易出錯。

流程圖



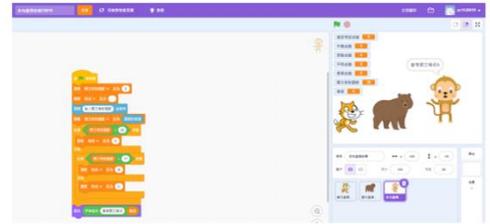
說明: Fig. 3-9 雙向選擇結構

說明: Fig. 3-10 多向選擇結構



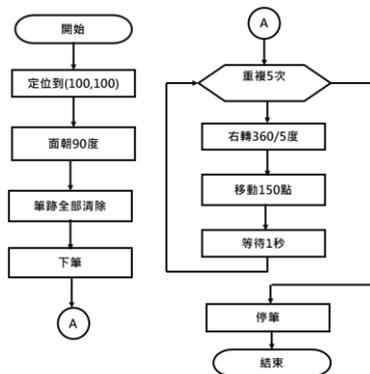
說明: Fig. 3-11 多向選擇結構

程式 (輸入36)



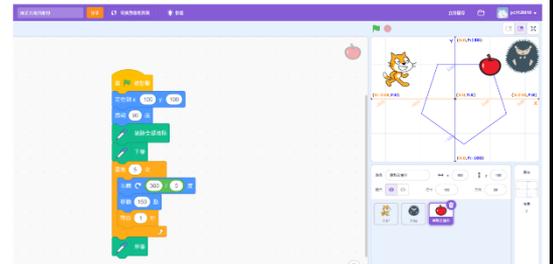
說明: Fig. 3-12 多向選擇結構

精熟題 - 流程圖
利用座標積木讓小貓畫出一個正五邊形

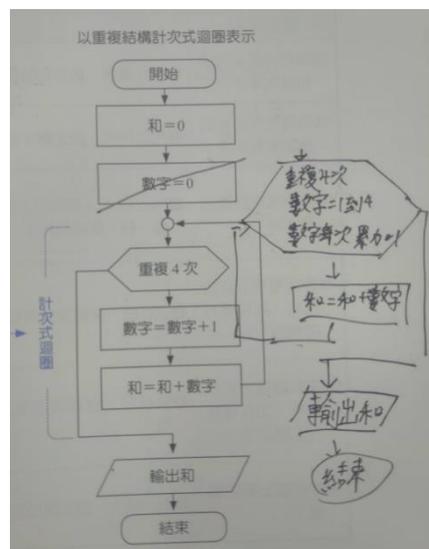


說明: Fig. 4-1 畫正多邊形

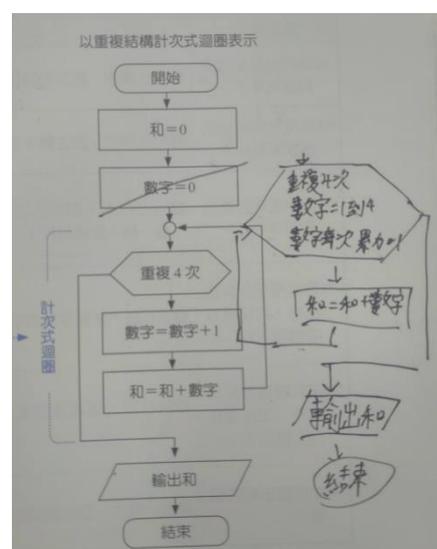
精熟題 - 程式
利用座標積木讓小貓畫出一個正五邊形



說明: Fig. 4-2 畫正多邊形



說明: Fig. 4-3 重複結構



說明: Fig. 4-4 重複結構

教學調整的脈絡

資訊科技課程有很大的一部分在學習程式設計，故而同學有一個觀念，認為資訊課就是電腦課，主要是學習如何寫程式(Scratch 程式設計)。殊不知，運算思維才是這門課的主軸，程式設計只是訓練運算思維的一種方式(目前公認的最好方式)。如同康軒課本第2頁標示的：**資訊科技是學習「以運算思維解決問題」的一門科目**

有誰來告訴學生這件事呢?老師會清楚告訴同學什麼是運算思維嗎?如何應用呢?如果只是寫出會動的程式，貌似解決問題的訓練，實則本末倒置。

所謂態度決定高度，能夠分析對策，最後落實為演算法，再以電腦為工具，解決問題才是正確的做法，經過這樣的訓練，期望學生的思維模式是 top down 的方式，日後不只會解決問題，更能發掘問題，定義問題。

演算法、流程圖、程式都有其架構及邏輯性，初學者必須 day one 得到正確的、精準的教導，尤其不能因為範例簡單而忽略這些方法論(methodology)，一旦錯誤的習慣養成，日積月累，根深蒂固，越來越難改，所謂失之毫釐謬以千里。

主要調整：

- 1 闡述運算思維、演算法、流程圖、程式設計之關聯。
- 2 流程圖的畫法，課本頗為混亂，根據流程圖繪製原則，統一畫法。
- 3 程式寫作，注重專案、角色、變數的命名，主要目的是要學生思考自己在做什麼。
- 4 修正課本關於初始化、迴圈架構不當之處，希望同學有個好的開始。
- 5 選擇結構補充單向選擇結構範例及多向選擇結構教學，以會考成績為例，引發學關心。
- 6 重複結構之計次迴圈則調整教學次序，先學習繪圖篇，不須連動變數，再學習計算篇，導入變數。
- 7 條件式迴圈則等計次迴圈教完再講解，捨棄複雜的密碼驗證範例，而以7下賽馬遊戲為範例。
- 8 以翰林課本為主兼容康軒及南一教科書。

成效分析

7年級的學生如同一張白紙，吸收力強，有待老師教導最基本的觀念，目前學到的，養成的，極有可能就帶進業界了。

希望，不會有同學說資訊課是防詐騙課，或資訊課是程式設計課，也希望減少同學寫出一些不合邏輯的程式。

至少，當他們不小心遺忘而犯錯，被指正時，會回憶起國中所學，不會振振有詞的說，以前老師都是這樣教的。

教學省思

程式設計教學往往流於片段，同學缺乏整合的能力，要求太多也易流於枯燥乏味，或可設計完成一個專案來彌補這個缺陷，有趣的內容，與能引起共鳴的作品，讓同學堅持程式技巧的熟練與整合，甚至 STEAM 的跨域整合，教學與學生學習都有正面的價值。

	<p>修正建議</p> <ol style="list-style-type: none"> 1. 同學使用資訊工具(登入、帳密管理、PPT、Scratch)的能力的差異，在7年級已漸呈現，加上融合教育下，同學學習方式各有不同，任務分級及不同教學平台的導入可以再加強(例如，導入 Egame 的打寇島)。 2. 學生實作的時間過長，進度難以掌握，應該來回巡視，給予適當鼓勵、提醒與建議。 3. 不同班級學習態度與能力仍有差異，教學時程較難掌握，需改善教材或教授方式。
<p>參考資料</p>	<ol style="list-style-type: none"> 1. 十二年國教課程綱要 2. 來尬冊數位教學平台 3. 翰林資訊科技1上備課用書 4. 康軒資訊科技1上備課用書 5. 南一資訊科技1上備課用書 6. BBC Learning WHAT IS AN ALGORITHM 7. 青蛙過河遊戲
<p>附錄</p>	