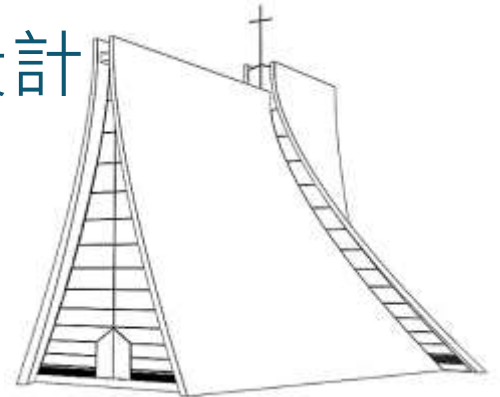




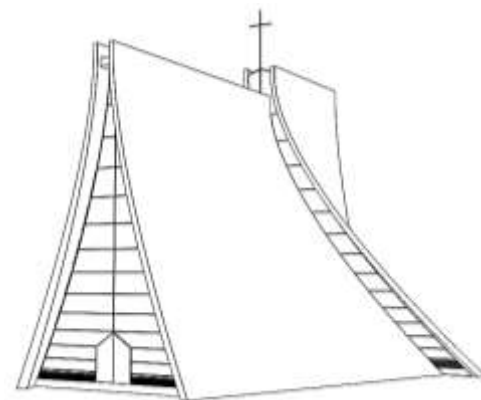
運算思維與程式設計

CH1-問題導向、運算思維與程式設計



非理工的我，

為什麼要學程式設計？





程式 + 跨域 = ∞

與程式共舞 — 黃翊 · 編舞家



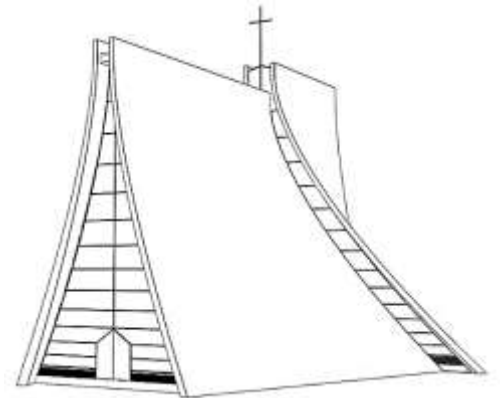


程式語言 = 第二外語



蘋果執行長庫克 (Tim Cook) 說，
學習編碼將比第二外語更加重要。

— 「讓你可以跟70億人交流」



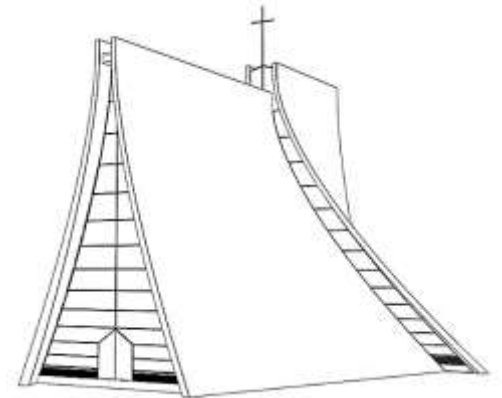


新全民運動，作夥來Coding！

教育部的108年課綱中，正式將數位素養、演算法和程式設計等能力指標納入國、高中課程。

未來，程式設計能力將成為國人具備的基本能力。

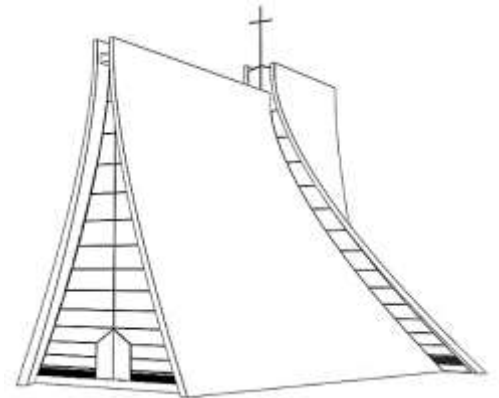
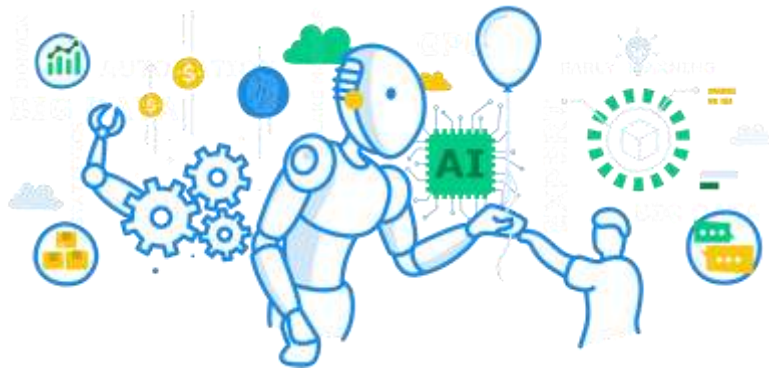
美國前總統歐巴馬 (Obama) 是位科技狂，他在任內曾親自示範寫程式，並編列了40億美元預算，將推廣電腦科學教育，視為國家戰略級的施政方針。





因應大智慧時代的來臨

因應各行各業的智慧化，舉凡智慧製造、智慧商業、智慧農業、智慧醫療、智慧城市等，各國爭相投入資源，培養各行各業人員具備程式撰寫的能力，我們怎能不加緊腳步！

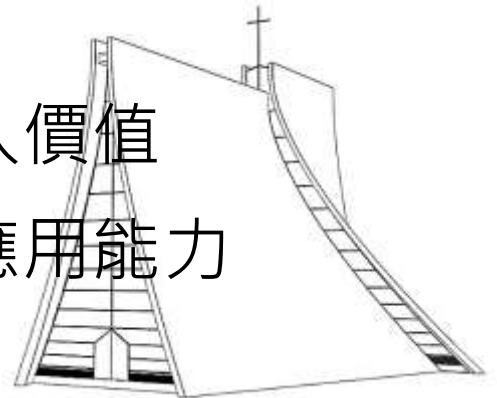




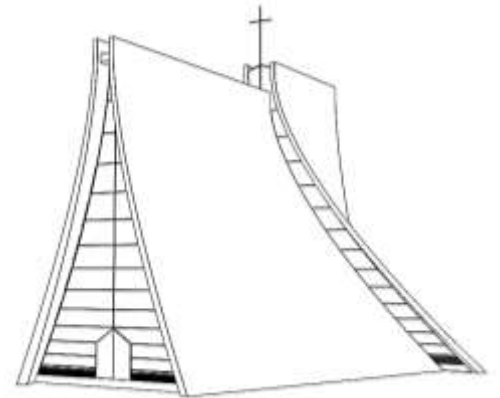
小 結

對於非理工科系的「你」，學習程式具有下列優勢：

- 1) 提升個人資訊素養與競爭力
- 2) 提升個人工作效益與生產力
- 3) 提升個人國際移動能力並創造個人價值
- 4) 提升個人跨領域數位創新與智慧應用能力



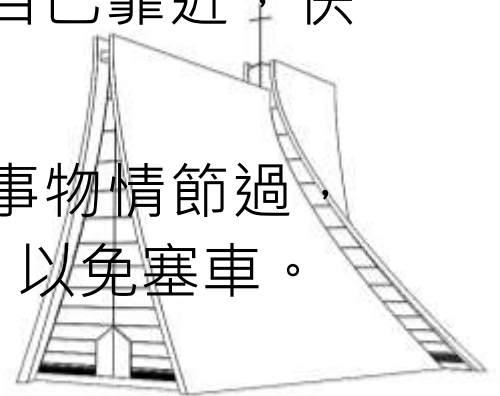
何謂運算思維？



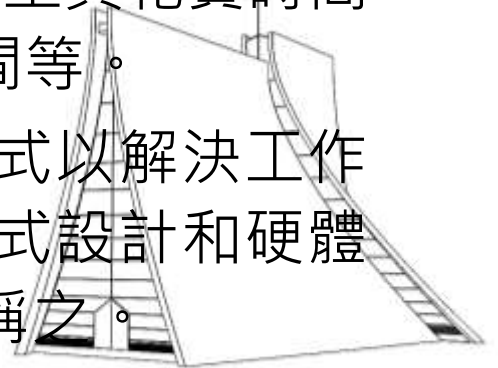
所謂「思維」，就是指一個人的處理事務和解決問題的方法，這些方法有直覺判斷、經驗導引、分析思考等由簡而繁的不同層次。

細分常見思維模式：

1. 「直覺思維」：依據視覺、聽覺、味覺、嗅覺、觸覺對事或物做出直接反應。例如看到車子向自己靠近，快速跳開。
2. 「經驗思維」：依據自己或他人發生過的事物情節過，作為行事的準則。例如下雨天提早出門，以免塞車。



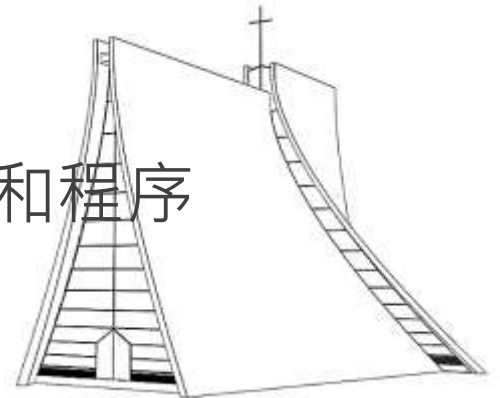
3. 「具體思維」：對特定個別的事與物之完整描述。例如搭乘100號公車從家裡到公司上班、家中寵物狗的模樣等。
4. 「抽象思維」：對同類型的事與物之概念認知。例如對所有犬類、自然萬物的概念和屬性的描述。
5. 「邏輯思維」：可逐步推理且符合常理的程序。例如南極就不應該有北極熊；要做一道菜，就需要有火、鍋子等。
6. 「分析思維」：瞭解問題的特質並分解成各種步驟且可解決的小問題。例如要決定如何從住家到公司，這個問題可分為 (a) 通勤方法，(b) 不同交通工具花費時間，(b) 路上交通狀況，(c) 距離公司上班時間等。
7. 「運算思維」：電腦是依賴工程師撰寫程式以解決工作上的問題，其思維模式必須符合軟體程式設計和硬體運作功能的方式，而這種思維模式即可稱之。



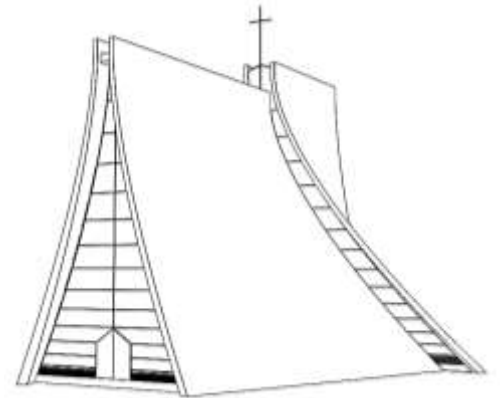


運算思維的關鍵步驟

- 1) **分解問題**：將一個大問題，拆解成許多的小問題
- 2) **模式識別**：尋找問題的相似度及前後關聯性，並予以歸類
- 3) **抽象思維**：化繁為簡，去掉多餘，專注核心
- 4) **邏輯推演 (演算法)**：
以流程圖方式，描述解決問題的步驟和程序



運算思維 VS. 程式設計





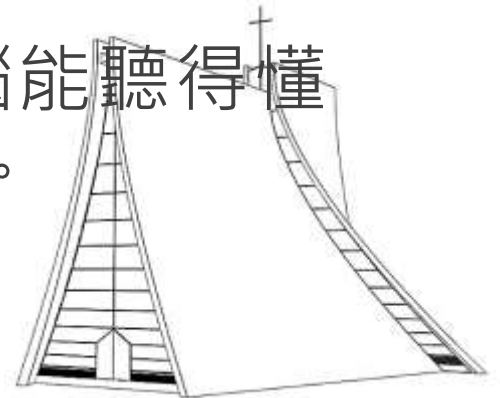
名詞解釋

「運算思維」 ★ 重視問題的解決方法與步驟

- 腦袋瓜所想，解一個問題的方法！

「程式設計」 ★ 實際編碼的過程

- 一種與電腦溝通的語言，讓電腦能聽得懂你所說的話，完成所交待的任務。





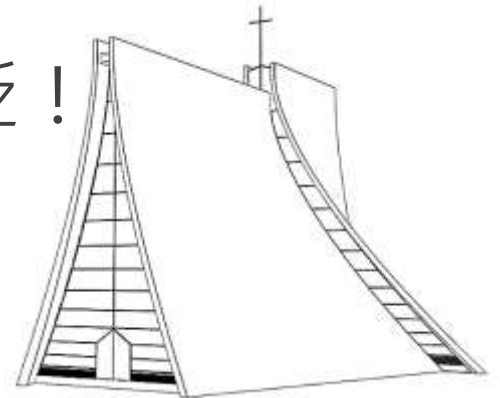
兩者關係

沒有**運算思維**，

程式設計將顯得雜亂無章。

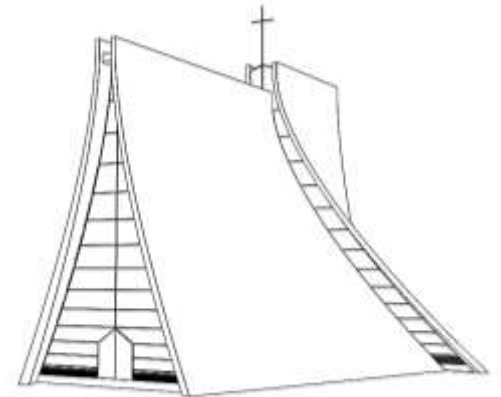
沒有**程式設計**，

運算思維將會變得很空泛！



問題導向學習

Problem-Based Learning (PBL)



問題導向學習是一種另類的教學方法，改變了過去僵化學習模式，在問題導向學習中，教師依據學習目標設計問題。

上課同學以分組方式—**協同學習**，彼此主動參與討論，發展解法並建立所需知識，教師僅從旁伴演促進者和引導者解色。





PBL 精神

自主學習



整合課程
知識

溝通互動



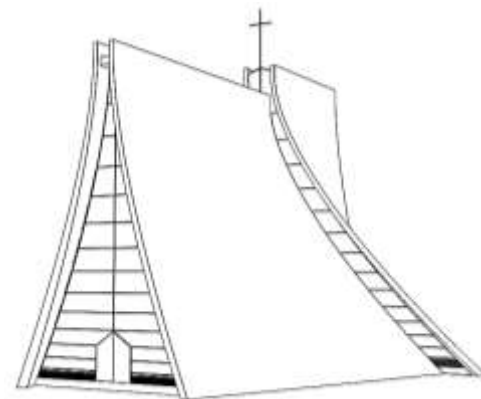
運用與
溝通表達

解決問題



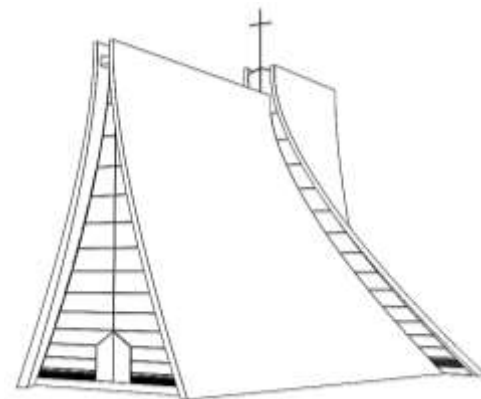
應用知識
與技能

程式語言的選用



那麼多種程式語言，


我該選擇哪一種？





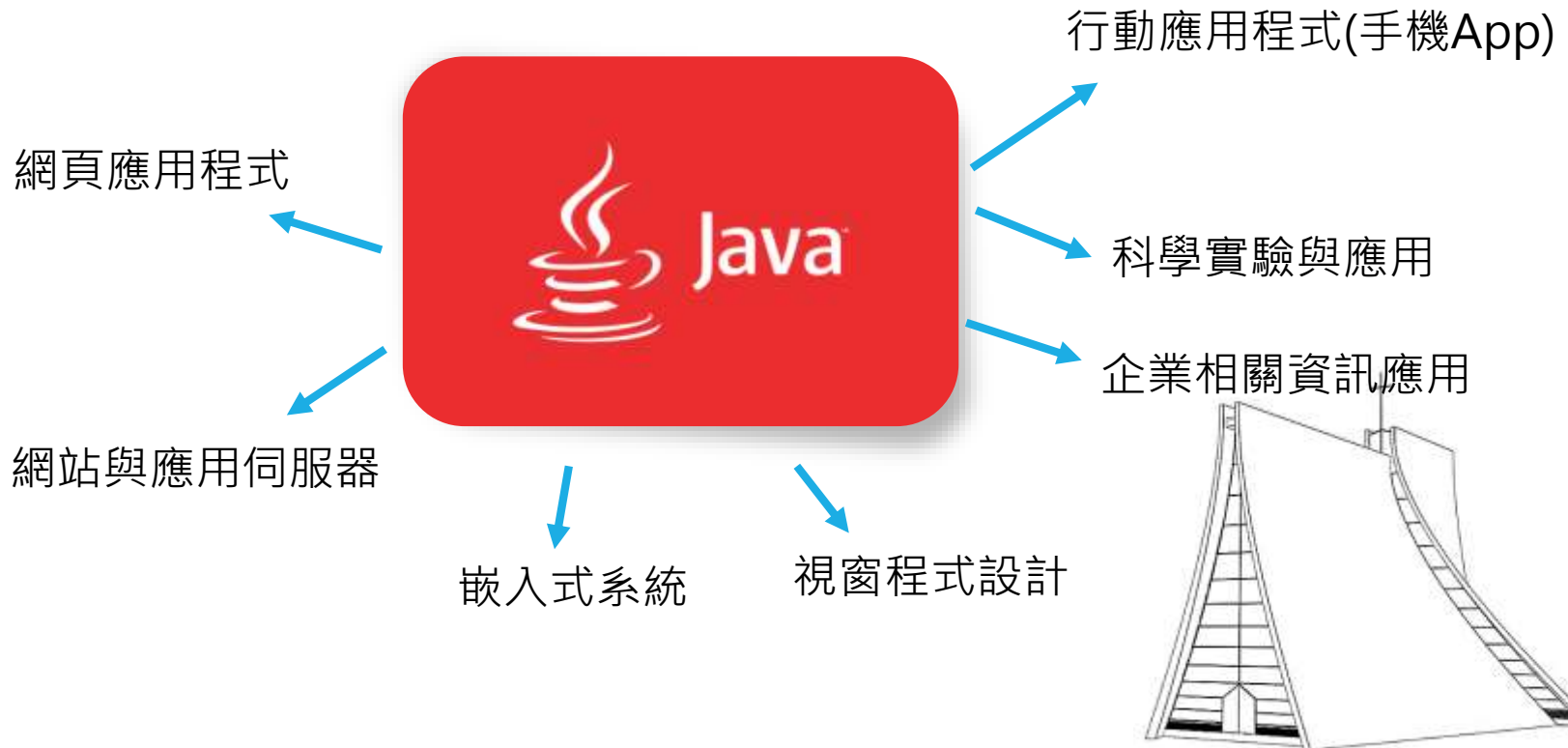
依熱門程度來看

The 2018 Top Programming Languages (IEEE spectrum)

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	99.7
3. Java	  	97.5
4. C	  	96.7
5. C#	  	89.4
6. PHP		84.9
7. R		82.9
8. JavaScript	 	82.6
9. Go	 	76.4
10. Assembly		74.1



從應用廣度來看



課程規劃

週次	標題		週次	標題	
1	問題導向、運算思維與程式設計	運算思維	9	運算子與運算式	基礎程式設計
2	問題定義、分解與識別		10	條件判斷	
3	抽象思考		11	迴圈控制	
4	邏輯推演		12	陣列與資料結構	
5	流程圖與CT2Flow	流程圖	13	亂數與API	
6	電腦與程式語言		14	電腦遊戲 – 剪刀、石頭、布	
7	變數與 CT2Code		15	電腦遊戲 – 終極密碼戰	
8	輸出與輸入處理		16	電腦遊戲 – 幾A幾B猜數字	



教學工具

Computational Thinking to Flowchart



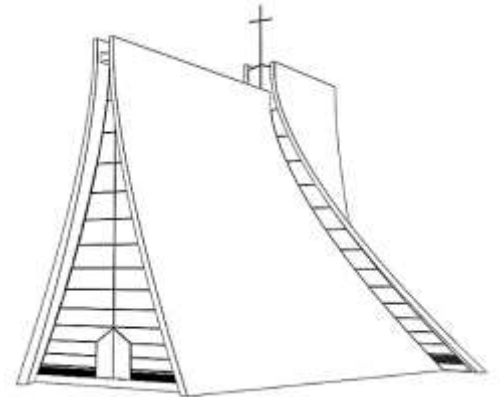
Computational Thinking to Code



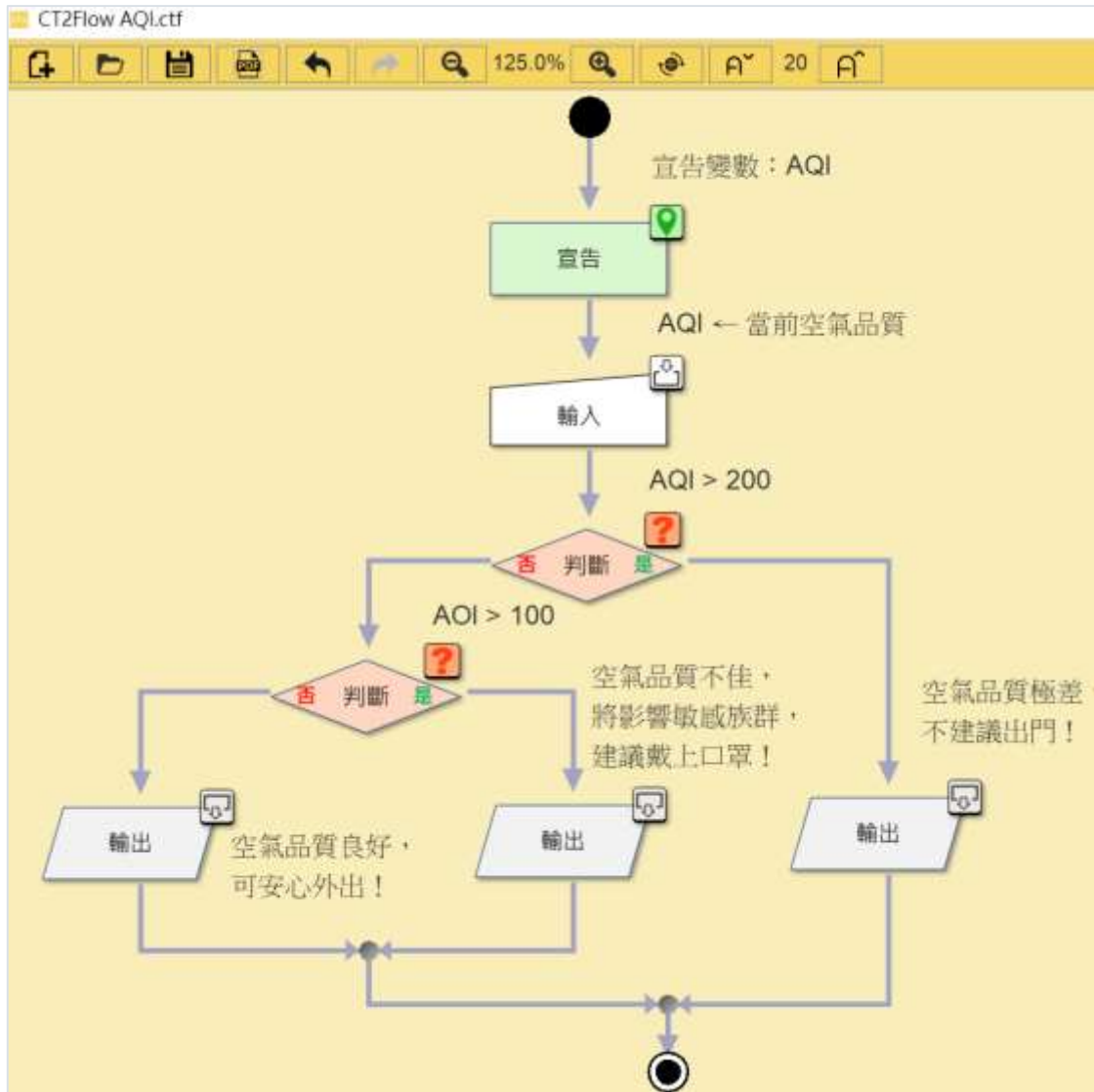


CT2Flow

CT2Flow是一套由東海大學團隊所開發的軟體工具，輔助學生能夠將解題的運算思維邏輯以流程圖搭配自然語言的方式呈現。



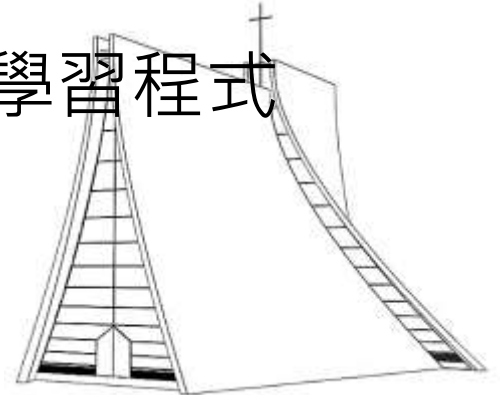
AQI	State
0 ~ 100	空氣品質良好
100 ~ 200	對敏感族群不健康
200 ~	危害，非常不健康

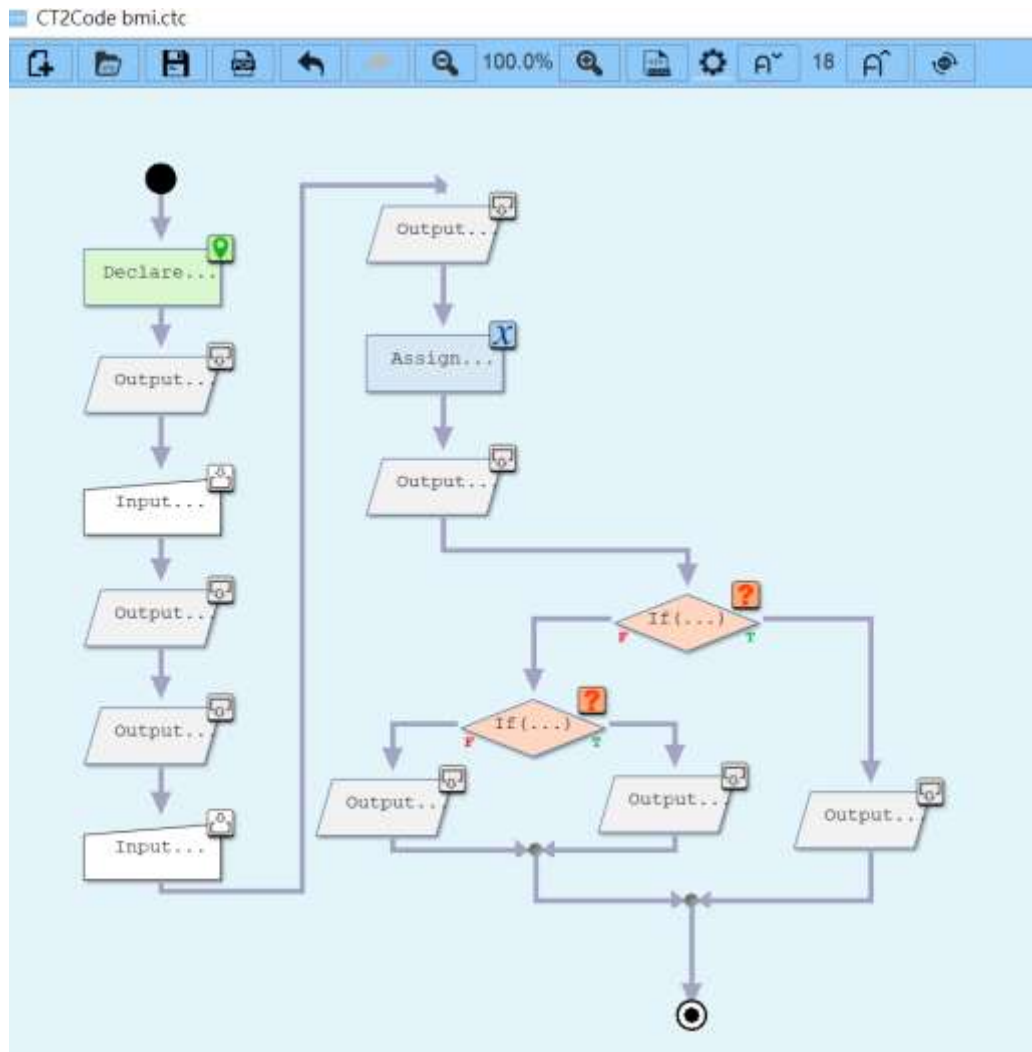




CT2Code

CT2Code是一套由東海大學團隊所開發的軟體工具，CT2Code可將抽象的計算思維與演算思維透過流程圖的方式呈現，並能依據流程圖產生對應的程式碼。期望大家能逐步學習程式設計，避免學習程式的挫折感。





```

15 System.out.println("你的 BMI 為：" + Math.round(bmi) );
16 if( bmi>24 ){
17     System.out.println("體位異常，請多留意!");
18 } else {
19     if( bmi<18.5 ){
20         System.out.println("體重過輕，請多留意!");
21     } else {
22         System.out.println("正常體位，繼續維持!");
23     }
24 }
25 sc.close();
26 }

```

CT2Code 啟動
開始執行程式
=====

身高 (公分) : 175.0 公分
體重 (公斤) : 80.0 公斤

你的 BMI 為 : 26
體位異常，請多留意!

送出

執行結果



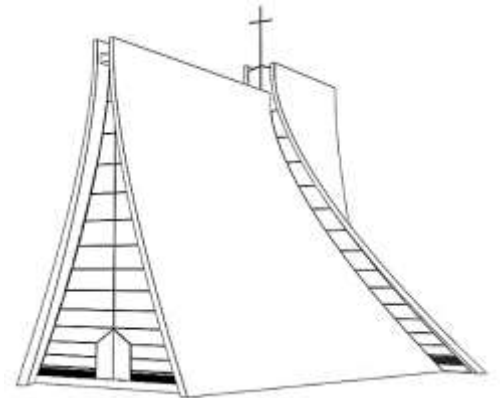
評分標準

1. 出席 / 平時成績：30%

- 「臨時」生病或有事的同學，需提前以mail方式跟我請假
- hsiung@thu.edu.tw
- 本課程**謝絕手機成隱症**的同學修課
- 例如：上課期間聊Line、玩手遊、看電影...

2. 課堂練習與作業：60%

3. 期末專案：10%



Thank You!