

Chapter 4：循序執行 X CAPSTONE 1

本章重點

編程不是一開始劈頭就寫你的想法，除非你寫過同樣的功能，第一步是理解問題，明確的定義問題後，再進行撰寫並在過程中修正錯誤。

準備材料



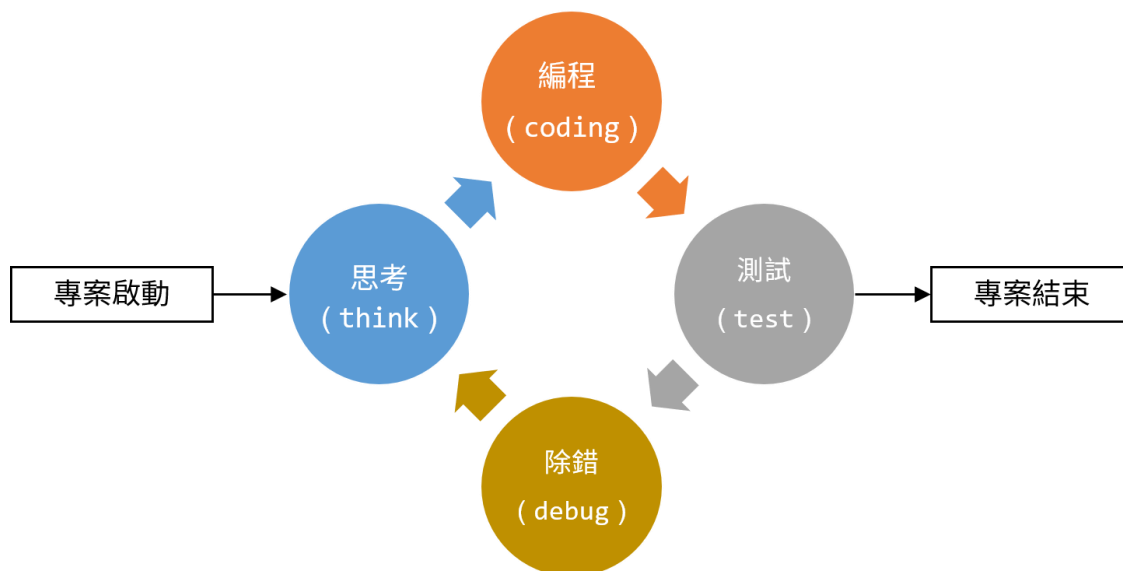
可上網的電腦

學習目標

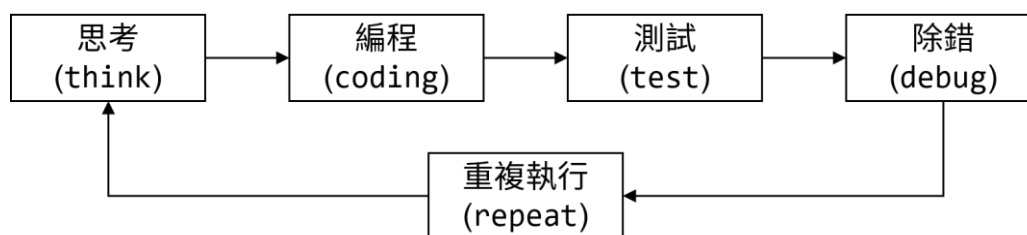
1. 學習標準程式設計的步驟。
2. 何謂可讀性高的程式碼。
3. 了解問題並採用標準設計方法，實作 CAPSTONE 1：將分鐘數轉換成幾小時幾分的轉換器。

4-1. 程式設計的步驟

就像閱讀書籍、算數學、彈鋼琴或打網球一樣，程式設計需要經過一個良好的學習計畫，搭配大量正確的練習方式。親自寫出每一行程式碼，與了解每一行。以下是程式設計的基本流程：



再次回顧下列的標準步驟，這是一個不斷重複的工作，直到程式達到最完美的設計。



- 編程前，應先思考問題本身的意義與目的，並釐清相關細節。
- 編程前，先列出所需要的輸入資訊以及可能的輸出資訊。
- 使用有意義的變數名稱並善用註解，以提高程式的可讀性。
- 程式的解答不只一種，找出最佳解是一項永不停歇的工作。

4-1-1. 思考階段的方法

思考階段：理解並釐清問題，並採用兩種藍圖先規劃設計。

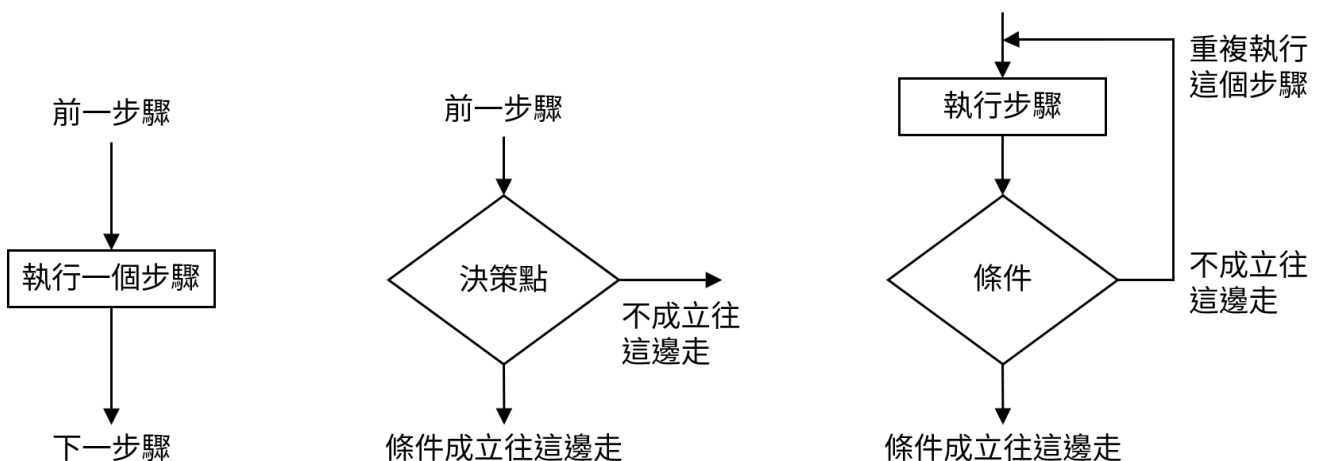
- 將解決問題的步驟視覺化：流程圖 (flowchart)。
- 將解決問題的步驟文字化：虛擬碼 (pseudocode)。

編程前，先確認以下事項：

- 要解決的問題是什麼？例如：我們想計算出一個圓形的面積。
- 這個程式需要跟使用者（user）互動嗎？例如：使用者輸入一個數字，經程式計算完成後，會顯示圓形面積。
- 輸入的資訊代表什麼意義？例如：輸入的數字代表一個圓半徑。
- 程式最後呈現資訊的格式是什麼呢？
 1. 簡易顯示，例如：程式可以只顯示 12.57 代表圓面積。
 2. 搭配文字來詳細說明，例如：顯示一段話："圓形的半徑是 2，面積是 12.57"。
 3. 以圖形顯示，例如：直接把圓形畫出來。
- 藉由思考、分析與詢問，將模糊的問題具體化。

將解決問題的步驟視覺化：流程圖

流程圖（flowchart）：圖形化的、明確定義程式的控制流程。



將解決問題的步驟文字化：虛擬碼

文字式的虛擬碼（pseudocode）：以口語文字來撰寫程式，再轉換成特定的程式碼。例如：計算出一個圓形的面積。

1. 要求使用者輸入圓形的半徑。
2. 利用公式算出圓形的面積。
3. 將算出的面積顯示出來。
4. 重複步驟 1 ~ 3，直到使用者輸入 "再見" 的條件為止。(重複工作為迴圈，參見 Chapter 7；判斷條件成立與否為判斷式，參見 Chapter 6)

4-1-2. 寫出可讀性高的程式

使用有意義的變數名稱

有意義的變數 (variable) 的名稱能夠幫助設計師，與其他的團隊人員管理程式碼。

- 難以閱讀的變數名稱：

```
a = 3.1
b = 2.2
c = a * b * b
```

- 明確好閱讀的變數名稱：

```
pi = 3.1
radius = 2.2
# 使用公式計算圓面積
circle_area = pi * radius * radius
```

將程式加上註解

註解可以幫助其他人了解這個程式。

- 較佳的註解：

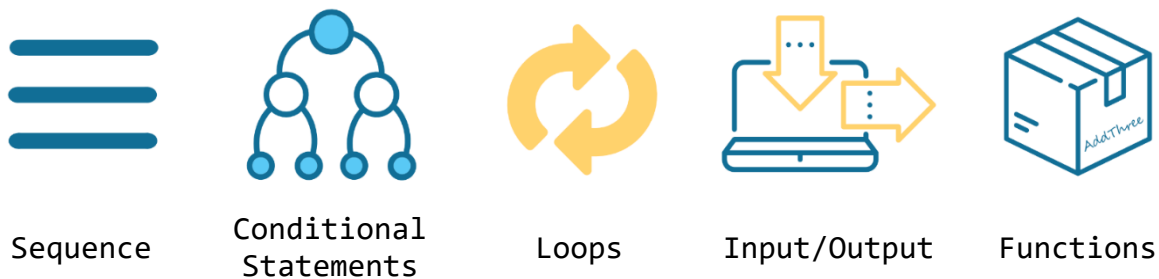
```
# 註解 1：使用公式計算圓面積
```

- 不好閱讀的註解：

```
# 註解 2：將圓周率乘半徑乘半徑
```

4-1-3. 程式的五大設計結構

所有的程式語言有五個主要的概念所組成，編程時應好好掌握這些概念，如圖所示：



1. 循序執行 (Sequence)：程式是一行接著一行，依序執行的。(參見 Chapter 3、4、5)
2. 條件敘述 (Conditional Statements)：有條件的執行某些敘述。(參見 Chapter 6)
3. 迴圈 (Loops)：有條件的重複執行某些敘述。(參見 Chapter 7)
4. 輸入/輸出 (Input/Output)：進行輸入與輸出的控制。
5. 自訂函式 (Functions)：將常常會重複使用的敘述，建立成函式來完成模組化。(本次課程未介紹)

4-2. 循序執行 - CAPSTONE 1：實作分鐘數轉換

循序執行 (Sequence) 是程式最基本的執行順序，一行接著一行，依序執行。本節將採用標準程式設計的步驟，實作出只有循序執行的分鐘數轉換。

4-2-1. 思考並釐清問題

第一步可先定義出完整的程式規格，並預期輸出的結果。

規格

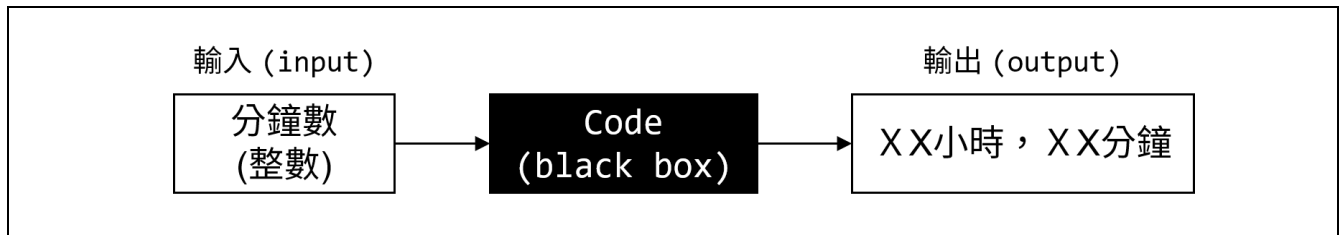
將分鐘數轉換為幾小時幾分。例如：輸入 123 分鐘，預期程式的輸出如下：

```
Hours
2
Minutes
3
```

先把程式碼當成黑盒子 (black box)，而不是一開始就寫程式碼，可以釐清思緒。

思考

可以先列出程式必要的輸入資料與輸出資料為何。



接下來可思考一些輸入與預期的輸出，稱之為**測試案例** (test case)。

思考

假設輸入值，預期輸出。

- 60 分鐘 → 1 小時 0 分鐘。
- 30 分鐘 → 0 小時 30 分鐘。
- 123 分鐘 → 2 小時 3 分鐘。

- 注意：學程式數學不用好，但用到時就要數學足夠好。

4-2-2. 先切分成小任務

切分成小任務 (task) 後組合，確認整體的實作與思維是否正確。

編程

目前有兩項 task 要處理，紀錄輸入值與顯示輸出值。

紀錄輸入值：取一個有意義的變數名稱，來記錄欲轉換的分鐘數，並轉換成數值型別。

```
minutes_to_convert = int(input('Enter minutes: '))
```

輸出計算結果：希望的輸出格式如下，<some number> 表示欲顯示的數值。

```
Hours  
<some number>  
Minutes  
<some number>
```

若決定使用電腦螢幕來輸出，可使用 `print()` 來顯示。

```
print('Hours')
print(hours_part)
print('Minutes')
print(minutes_part)
```

4-2-3. 實作轉換公式

切割成小任務，幫助思考

切分成小任務 (task) 後組合，確認整體的實作與思維是否正確。

- 123 分鐘 / 60 → 2.05。 (不符合規格)
- 123 分鐘 → 2 小時 3 分鐘。 (符合)

思考

有兩項 task 要處理，轉換小時數與分鐘數。

找出小時數：123 分鐘 / 60 → 2.05，整數位代表小時，符合預期。

編程

把輸入值除以 60 後，轉換成 `int`，便得到小時數。

```
hours_decimal = minutes_to_convert / 60
hours_part = int(hours_decimal)
```

1. $123 / 60 = 2.05$ ，分鐘數 = $0.05 * 60$ 。
2. 使用餘數計算， $123 \% 60 = 3$ 。

組合程式碼

4-1

採用第一種分鐘數解法的程式。

```
minutes_to_convert = int(input('Enter minutes: ')) # 輸入欲轉換的分鐘數

hours_decimal = minutes_to_convert / 60 # 分鐘數轉換為小數，例如：2.05
hours_part = int(hours_decimal) # 取整數部分為小時，例如：int(2.05) = 2

minutes_decimal = hours_decimal - hours_part # 取得小數點部分
minutes_part = round(minutes_decimal * 60) # 將小數點部分還原分鐘數

print('Hours')
print(hours_part)
print('Minutes')
print(minutes_part)
```

由上面程式必須知道的幾點：

1. 執行程式遇到 `input()` 時，程式會暫停，直到使用者輸入完成，並按下 `Enter`。
2. `input()` 回傳的是字串，後面的計算需要數值型別，所以再加上 `int()` 做型別轉換。
3. 這是一個循序執行的程式，執行到最後一程式便會結束。
4. `#` 的註解不宜太多，這裡是為了方便教學說明。
5. 只有 `print()` 才會在螢幕輸出資料。

```
"""
    4-2-clock converter.py
    -----
    Creates a clock converter.
    The converter transfers minutes to hours part and minutes part.
"""

minutes_to_convert = int(input('Enter minutes: '))

# 轉換小時數
hours_decimal = minutes_to_convert / 60
hours_part = int(hours_decimal)

# 轉換分鐘數
minutes_part = minutes_to_convert % 60

print('Hours')
print(hours_part)
print('Minutes')
print(minutes_part)
```

4-2 是一個完整的程式，包含較佳的註解：

1. """ 4-2... """ 為多行註解，程式不會執行，主要是說明程式的檔名與目的。
2. 可將部分程式碼的註解寫在上方。例如：# 轉換小時數，並適當的空格區隔下一個功能程式碼。
3. 註解為英文或中文皆可，英文註解為較佳。

練習 4-2

1. 寫一個程式，變數初始值為 75，代表華氏溫度，使用公式 $c = (f - 32) / 1.8$ ，將華氏溫度轉成攝氏溫度，並顯示計算結果。

2. 寫一個程式，讓使用者可輸入一個數值，例如：5，代表英哩 (miles)。

公里 (km) = 英哩 (miles) / 0.62137

公尺 (meters) = 1000 * 公里 (km)

將英哩 (miles) 轉為公里 (km) 及公尺 (meters) 並顯示以下的結果：

```
Enter miles: 5
Km
8.046735439432222
Meters
8046.735439432222
```