

# 程式設計 (**Programming**)

真理大學 資訊工程系 吳汶涓老師

CH03

結構化程式的開發



# 本章綱要

3-1 簡介

3-2 演算法

3-3 虛擬程式碼

3-4 控制結構

3-5 if 選擇敘述式

3-6 if...else 選擇敘述式

3-7 重複敘述式

3-8 案例研究1

3-9 案例研究 2

3-10 案例研究3

3-11 指定運算子

3-12 遞增和遞減運算子

## 3.1 簡介

### ■ 在撰寫程式之前:

- 先充分**了解問題**
- 仔細規劃**解決**這個問題的**方法**

演算法  
Algorithm

### ■ 在撰寫程式時:

- 了解有哪些「建構區塊」可供使用
- 遵從良好的程式設計原則



#### **良好的程式設計習慣 2.5**

請選用具有意義的變數名稱，來增進程式本身的可讀性。



#### **常見的程式設計錯誤 2.14**

忘了在 scanf 敘述式內的變數之前加上 & (若此變數應該加上 &)。

## 3.2 演算法

### ■ 計算問題

- 可歸納成以**特定的順序**來執行一系列的動作

### ■ 演算法(algorithm)：可視為以下**程序(procedure)**

- 執行的**動作 (actions)**
- 執行這些動作的**順序 (order)**

### ■ 執行控制 (program control)

- 在電腦程式裡指定敘述式執行的順序

■ 例如：「一大早起床後所該做的事」

- (1) 起床
- (2) 脫掉睡衣
- (3) 洗澡
- (4) 穿好衣服
- (5) 吃早餐
- (6) 開車上班

- (1) 起床
- (2) 脫掉睡衣
- (3) 穿好衣服
- (4) 洗澡
- (5) 吃早餐
- (6) 開車上班



## 3.3 虛擬程式碼

### ■ 虛擬程式碼 (Pseudocode)

- 給人看的非正規的語言，用來幫助你發展演算法
- 類似日常生活的**英文**
- **並不能**在實際的電腦上**執行**
- 幫助我們在撰寫程式之前，先思考程式應如何撰寫
  - 容易轉換成相對應的C語言程式
  - 由可執行的敘述式所組成



```
If student's grade is greater than or equal to 60
    Print "Passed"
else
    Print "Failed"
```

## 3.4 控制結構

- 循序式的執行 (sequential execution)
  - 敘述式是以他們在程式中的順序一個接一個地被執行
- 控制權的移轉 (transfer of control)
  - 下一個執行的敘述式並非依照順序的
  - 過度使用goto敘述式會產生很多問題
- 所有程式均可由三種控制結構寫成
  - 循序結構：C內建的特性，程式預設是循序執行的
  - 選擇結構：C具有三種類型(if, if ... else, switch)
  - 重複結構：C具有三種類型(while, do ... while, for)

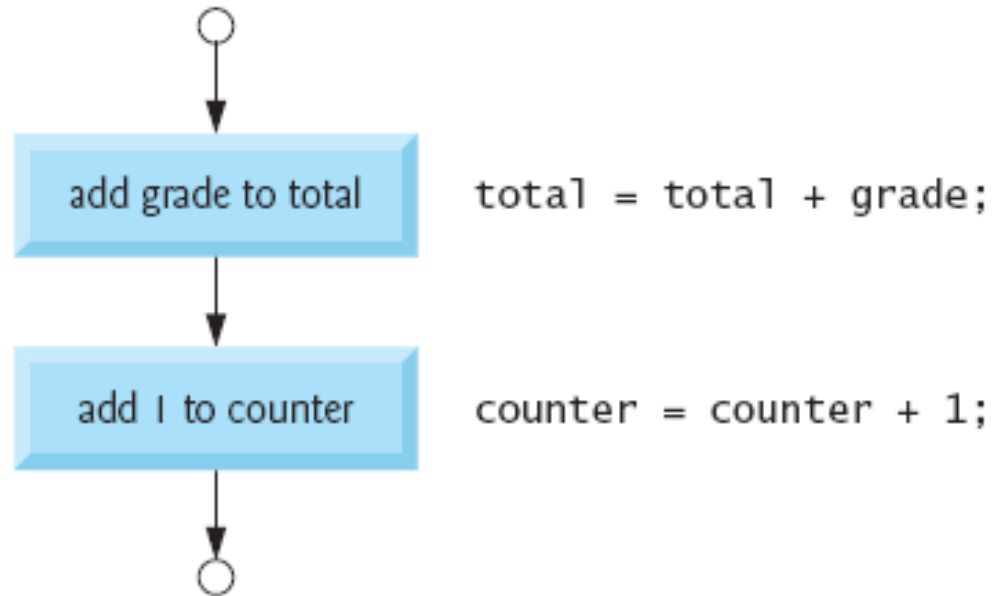


圖 3.1 C 循環結構的流程圖



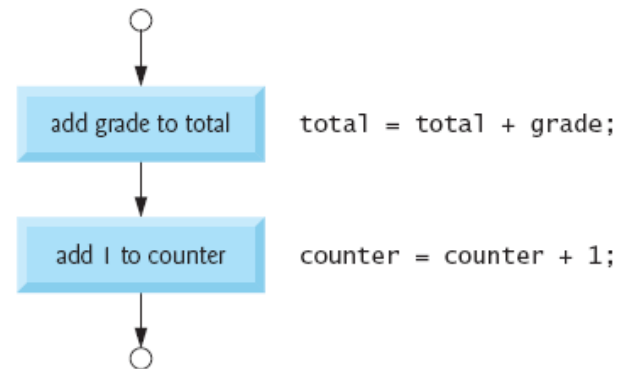


圖 3.1 C 循環結構的流程圖

## ■ 流程圖

- 演算法的一種圖形表示方式
- 使用具有特殊涵義的標誌來繪製，並用**流向**的箭頭連接起來
  - **矩形：動作符號**
  - **菱形：判斷符號**
  - **橢圓形：開始或結束符號**

## ■ 單一入口/單一出口

- 亦即**循序結構**
- 將一個控制結構的入口連接到另一控制結構的出口
- 可使程式的建構更為容易

# 練習

- 請撰寫一個程式讀取圓的半徑(整數)，然後印出圓的直徑、圓周長和面積的資訊。圓周率為 **3.14159**，記得輸出時，使用 **%f** 的轉換指定詞。
  - 請先試著繪出碁流程圖
  - 依據流程圖，撰寫對應C程式碼



# 練習

- 撰寫一個程式計算身體質量指數(BMI)，計算公式如下，其中體重(公斤)與身高(公尺)由使用者輸入，可輸入小數值，輸出時使用%f 的轉換指定詞。

$$\text{BMI} = \frac{\text{體重(公斤)}}{\text{身高(公尺)} * \text{身高(公尺)}}$$



## 3.5 if 選擇敘述式

### ■ 選擇結構

- 用來選取不同功能的動作
- 虛擬程式碼：

*If student's grade is greater than or equal to 60  
Print "Passed"*

- 條件為**true**，才會執行print敘述式；條件為**false**，不會執行
- 菱形符號 (判斷符號)：包含一個運算式(真或偽)與適當的流向作選擇
- 將虛擬碼轉換為C語言

```
if ( grade >= 60 ) {  
    printf( "Passed\n" );  
}
```

- 使用 tab 縮排會使程式易於閱讀 (可讀性)

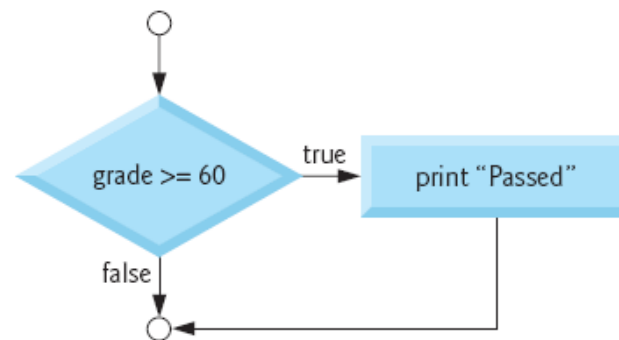


圖 3.2 單一選擇 if 敘述式的流程圖

# 練習

- 撰寫一個程式讀取學生的成績，判斷是否及格。



## 3.6 if...else 選擇敘述式

### ■ if

- 只執行條件為真時的敘述動作

### ■ if ... else

- 當條件為真或偽時，分別執行各自指定動作
- 虛擬碼：

*If student's grade is greater than or equal to 60*

*Print "Passed"*

*else*

*Print "Failed"*

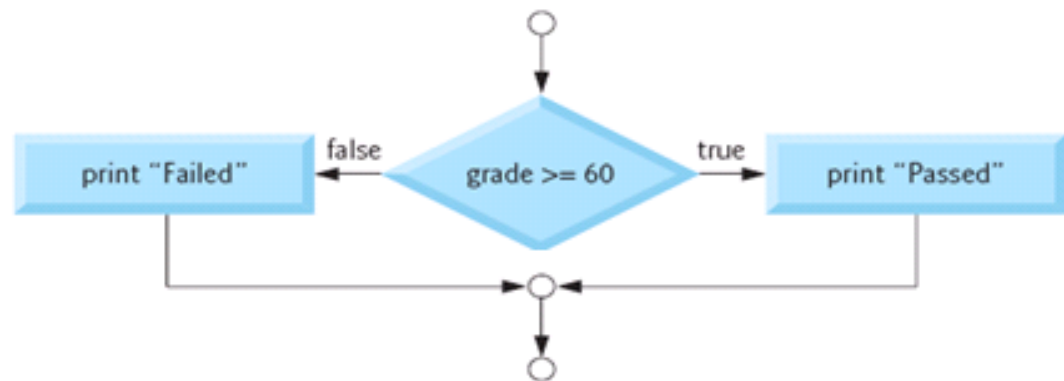


圖 3.3 C 之雙重選擇 if...else 敘述式的流程圖

- ❑ 虛擬碼：  
*If student's grade is greater than or equal to 60*  
*Print "Passed"*  
*else*  
*Print "Failed"*

- ❑ C程式碼：

```
if ( grade >= 60 )  
    printf( "Passed\n");  
else  
    printf( "Failed\n");
```

- ❑ 注意空白與縮排的使用

## ■ 三元條件運算子(?:)

- ❑ 使用了三個運算元(含條件、條件為真的運算式與條件為偽的運算式)
- ❑ `Printf("%s\n", grade >= 60 ? "Passed":"Failed");`

# 練習

- 撰寫一個程式讀取使用者輸入的兩個數值(為整數、小數)，判斷哪一個數值比較大。
- 撰寫一個程式讀取使用者輸入的整數數值，判斷其數值是為偶數或奇數。
  - 使用三元條件運算子 (?:)





## ■ 複合敘述式 (compound statement)

- 被包在大括號裡的一組敘述式，又稱為一個區塊(block)
- 範例：

```
if ( grade >= 60 ) {  
    printf( "Passed.\n" );  
}  
else {  
    printf( "Failed.\n" );  
    printf( "You must take this course again.\n" );  
}
```

- 如果else區塊沒有用大括號包起來，則成績大於等於60分也會印出You must take this course again.



### 軟體工程的觀點 3.1

複合敘述式可放在程式中任何可放單一敘述式的地方。



### 常見的程式設計錯誤 3.1

忘了用大括號將複合敘述式包起來。



### 常見的程式設計錯誤 3.2

在 if 敘述式的條件式之後放置一個分號，像是「`if (grade >= 60);`」將會使單一 選擇的 if 敘述式產生邏輯錯誤

## ■ 注意

- **語法錯誤**(syntax error)：會在編譯時產生，由編譯器發現
- **邏輯錯誤**：會在執行時期造成影響
  - 致命的：使得程式失敗並提早終止。
  - 非致命：可讓程式繼續執行，但會產生不正確的執行結果

## ■ 巢狀的if...else敘述式

- ❑ 將if...else敘述放到另一個if...else敘述裡，用以檢測多重的狀況
- ❑ 一旦滿足了某個條件式，就會跳過剩餘的敘述式
- ❑ 通常不建議使用過深的縮排

❑ 虛擬碼：

```
If student's grade is greater than or equal to 90  
  Print "A"  
else  
  If student's grade is greater than or equal to 80  
    Print "B"  
  else  
    If student's grade is greater than or equal to 70  
      Print "C"  
    else  
      If student's grade is greater than or equal to 60  
        Print "D"  
      else  
        Print "F"
```

□ C程式碼：

```
if ( grade >= 90 )
    printf( "A\n" );
else
{
    if ( grade >= 80 )
        printf("B\n");
    else
    {
        if ( grade >= 70 )
            printf("C\n");
        else
        {
            if ( grade >= 60 )
                printf( "D\n" );
            else
                printf( "F\n" );
        }
    }
}
```

可改寫



```
if ( grade >= 90 )
    printf( "A\n" );
else if ( grade >= 80 )
    printf( "B\n" );
else if ( grade >= 70 )
    printf( "C\n" );
else if ( grade >= 60 )
    printf( "D\n" );
else
    printf( "F\n" );
```

簡潔、高可讀性

# 練習

- 撰寫一個購物程式，自行設定價格，並顯示應付金額。若價格超過**5000元 (含)** 以上，則打**7折**，若是**4000~4999元**之間，打**8折**，若是**3000~3999元**之間，打**9折**，**3000元(不含)**以下不打折。



# 練習

- 撰寫一個程式，由使用者輸入三個整數，判斷這三個整數的最大值與最小值，並輸出其結果。

