

# 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

## 3.7 while 重複敘述式

### ■ 重複結構 (repetition)

- 指定在某種條件**持續為真**時，重複執行同一項動作，直到條件為假
- 演算法：

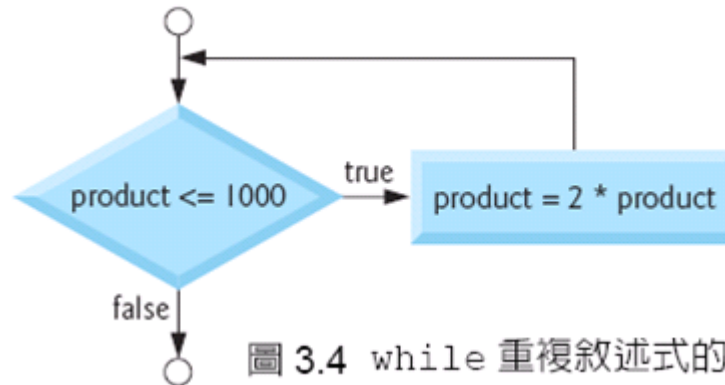


圖 3.4 while 重複敘述式的流程圖

- C程式碼：

```
int product = 2;
while ( product <= 1000 )
    product = 2 * product;
```

**Q: product = ?**



### 常見的程式設計錯誤 3.3

在 while 結構的本體內，沒有任何一個動作能讓 while 的條件變成偽。通常這種重複結構將不會停止——此種錯誤稱為「無窮迴圈」。

Example:

```
int product = 2;  
while( product !=1000 )  
    product = 2 * product;
```

Example:

```
int i = 1, sum = 0;  
while( i <=10 ){  
    sum = sum + i;  
}
```

# 練習

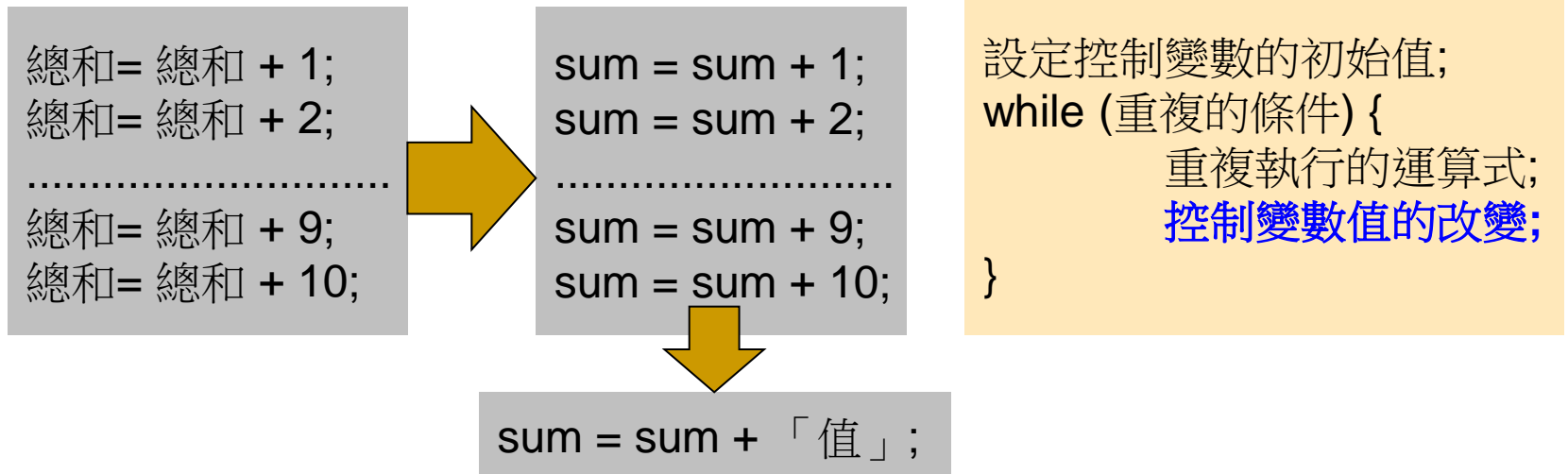
- 撰寫一個程式計算並輸出2的次方數，即 2, 4, 8, 16, 32, 64, ...，直到值超過2000則停止輸出。
  - 將上述的2，改為由使用者輸入之



## 3.8 while 案例研究1：計數器控制

### ■ 計數器控制的重複結構

- 計數器用來控制執行的次數 (用於**已知明確的重複次數**)
- 迴圈會一直重複，直到計數器等於某個值
- 示範重複敘述的最佳題材應該是**數學的累加計算**  
試想，計算 $1+2+\dots+9+10$ 的數學式，可以分階段來計算：



□ 範例：計算一個班級內十個學生的測驗平均分數

---

```
1  Set total to zero
2  Set grade counter to one
3
4  While grade counter is less than or equal to ten
5      Input the next grade
6      Add the grade into the total
7      Add one to the grade counter
8
9  Set the class average to the total divided by ten
10 Print the class average
```

---

圖 3.5 利用計數器控制的重複結構來解決全班平均問題的虛擬碼演算法

```

1  /* Fig. 3.6: fig03_06.c */
2
3  #include <stdio.h>
4
5
6  int main( void )
7  {
8      int counter;
9      int grade;
10     int total;
11     int average;
12
13     /* initialization phase */
14     total = 0;
15     counter = 1;
16
17     /* processing phase */
18     while ( counter <= 10 ) {
19         printf( "Enter grade: " );
20         scanf( "%d", &grade );
21         total = total + grade;
22         counter = counter + 1;
23     } /* end while */

```

計數器

累計總和的變數

設定初始值

遞增計數器

圖 3.6 以計數器控制重複結構來解決全班平均問題的 C 程式及其執行範例



```
24
25  /* termination phase */
26  average = total / 10;
27
28  printf( "Class average is %d\n", average );
29  return 0;
30 }
```

計算平均值，產生一個整數值

```
Enter grade: 98
Enter grade: 76
Enter grade: 71
Enter grade: 87
Enter grade: 83
Enter grade: 90
Enter grade: 57
Enter grade: 79
Enter grade: 82
Enter grade: 94
Class average is 81
```

圖 3.6 以計數器控制重複結構來解決全班平均問題的 C 程式及其執行範例



### 常見的程式設計錯誤 3.5

如果 counter 或 total 沒有設定初值，這樣子程式執行的結果將可能不正確。這是個邏輯錯誤的例子。

# HW3 作業

- 撰寫一個程式計算一個非負整數 $n$ 的階乘值。
  - $n$  的階乘為  $n! = n * (n-1) * (n-2) * \dots * 1$   
 $= 1 * 2 * 3 * \dots * (n-1) * n$
  - 執行累乘的動作，最後輸出最後的階乘值



## 3.9 while 案例研究2：警示訊號控制

### ■ 問題發展變成

- 一個求全班平均的程式，它可以處理任何數目的成績，但是，**不知道班上學生人數**
- 程式要如何知道何時應該結束

### ■ 利用**警示值**

- 也稱為**信號值 signal value**或**旗標值 flag value**
- 代表「資料輸入的結束」
- 使用者鍵入警示值的話，迴圈便將終止
- 警示值必須是個不能與合法資料值互相混淆的數 (**通常使用 -1**)

## ■ 從上而下逐步改進

- 先從總敘述式的虛擬碼開始：

*Determine the class average for the quiz*

- 再分割成數項較小的工作，並依執行的順序列出：

*Initialize variables*

*Input, sum and count the quiz grades*

*Calculate and print the class average*

## ■ 大多數的程式都可分成三個階段：

- 初始化 (initialization)：設定程式變數的初值
- 處理 (processing)：輸入資料並修改相對應的變數
- 結束 (termination)：計算並輸出最後的結果

- Initialize variables 改進為

  - Initialize total to zero*

  - Initialize counter to zero*

- Input, sum and count the quiz grades 改進為

  - Input the first grade (possibly the sentinel)*

  - While the user has not as yet entered the sentinel*

    - Add this grade into the running total*

    - Add one to the grade counter*

    - Input the next grade (possibly the sentinel)*

- Calculate and print the class average 改進為

  - If the counter is not equal to zero*

    - Set the average to the total divided by the counter*

    - Print the average*

  - else*

    - Print "No grades were entered"*



### 軟體工程的觀點 3.3

總敘述式及每一次的改進都是一份完整的演算法，不同的只是詳細的程度。

*Determine the class average for the quiz*



*Initialize variables  
Input, sum and count the quiz grades  
Calculate and print the class average*



*Initialize total to zero*

*Initialize counter to zero*

*Input the first grade (possibly the sentinel)*

*While the user has not as yet entered the sentinel*

*Add this grade into the running total*

*Add one to the grade counter*

*Input the next grade (possibly the sentinel)*

*If the counter is not equal to zero*

*Set the average to the total divided by the counter*

*Print the average*

*else*

*Print "No grades were entered"*

## ■ 注意

- ❑ 虛擬碼中，while內的敘述式用縮排表示屬於while的
- ❑ 虛擬程式碼只是用來幫助發展程式
- ❑ **除以零**是一種致命錯誤，會使程式當掉(crashing)

---

```
1  Initialize total to zero
2  Initialize counter to zero
3
4  Input the first grade
5  While the user has not as yet entered the sentinel
6      Add this grade into the running total
7      Add one to the grade counter
8      Input the next grade (possibly the sentinel)
9
10 If the counter is not equal to zero
11     Set the average to the total divided by the counter
12     Print the average
13 else
14     Print "No grades were entered"
```

圖 3.7 利用警示訊號控制的重複結構來解決全班平均問題的虛擬碼演算法

```
1  /* Fig. 3.8: fig03_08.c */
2
3  #include <stdio.h>
4
5
6  int main( void )
7  {
8      int counter;
9      int grade;
10     int total;
11
12     float average;
13
14     /* initialization phase */
15     total = 0;
16     counter = 0;
17
18     /* processing phase */
19     /* get first grade from user */
20     printf( "Enter grade, -1 to end: " );
21     scanf( "%d", &grade );
22
```

浮點數：非整數




圖 3.8 以警示值控制重複結構來解決全班平均問題的 C 程式及其執行範例



23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45

```
while ( grade != -1 ) {  
    total = total + grade;  
    counter = counter + 1;  
  
    printf( "Enter grade, -1 to end: " );  
    scanf("%d", &grade);  
}
```

迴圈會不斷重複，  
直到使用者輸入  
-1為止

```
if ( counter != 0 ) {  
  
    average = ( float ) total / counter; /* avoid truncation */  
  
    printf( "Class average is %.2f\n", average );  
}  
else {  
    printf( "No grades were entered\n" );  
}
```

避免除以零

強制轉型

圖 3.8 以警示值控制重複結構來解決全班平均問題的 C 程式及其執行範例

```
46
47     return 0; /* indicate program ended successfully */
48 } /* end function main */
```

```
Enter grade, -1 to end: 75
Enter grade, -1 to end: 94
Enter grade, -1 to end: 97
Enter grade, -1 to end: 88
Enter grade, -1 to end: 70
Enter grade, -1 to end: 64
Enter grade, -1 to end: 83
Enter grade, -1 to end: 89
Enter grade, -1 to end: -1
Class average is 82.50
```

```
Enter grade, -1 to end: -1
No grades were entered
```

圖 3.8 以警示值控制重複結構來解決全班平均問題的 C 程式及其執行範例

- 假使，**while**迴圈沒有大括號，則電腦將會誤認：

```
while ( grade != -1 )  
    total = total + grade;  
counter = counter + 1;  
printf( "Enter grade, -1 to end: " );  
scanf( "%d", &grade );
```

程式將會造成無窮迴圈

- 控制輸出浮點數值的精準度

- printf轉換指定詞%.2f是用來控制輸出資料為兩個小數點的格式
- %.f是預設精準度，有六個小數點
- 其值會被四捨五入成所指定的精準度，但在記憶體內的值未被改變

```
printf( "%.2f\n", 3.446); /* prints 3.45 */  
printf( "%.1f\n", 3.446); /* prints 3.4 */
```



### 常見的程式設計錯誤 3.8

在 `scanf` 敘述式之格式控制字串內使用具有精準度的轉換指定詞是不正確的。精準度只能使用在 `printf` 的轉換指定上。

```
21    scanf( "%d", &grade );           /* read grade from user */
22
40    /* display average with two digits of precision */
41    printf( "Class average is %.2f\n", average );
```

**錯誤：**`scanf("%.2f", &grade);`



### 測試和除錯的小技巧 3.3

不要比較兩個浮點數的相等性。

# 練習

- 請找出並更正下列各項的錯誤：

```
(a) while(c <= 5){  
    product = product * c;  
    c = c + 1;
```

```
(b) scanf("%.4f", &value);
```

```
(c) if(gender == 1)  
    printf("woman \n");  
else;  
    printf("man \n");
```

# 練習

- 請問這個程式會印出什麼？

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int count = 1;
    while(count <= 10){
        printf("%s\n", count%2 ? "*****" : "+++++++");
        count=count+1;
    }
    system("pause");
    return 0;
}
```

# 練習

- 撰寫一個程式計算每公升汽油所跑的公里數，由使用者輸入每次的公里數及所加入的汽油公升數
  - 參考課本 pp.3-43, ex. 3.16
  - $\text{每公升跑多少公里} = \text{公里數} / \text{公升數}$
  - 此程式可由使用者不斷輸入，直到輸入為 -1 為止

