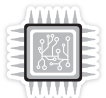


04

運算式和運算子

交互閃爍LED燈





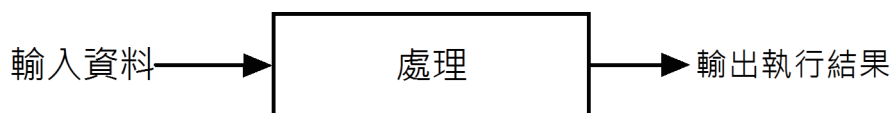
4-1 認識運算式和運算子

程式語言的運算式 (expressions) 是一個執行運算的程式敘述，可以產生運算結果的常數值，整個運算式可以簡單到只有單一常數值或變數，或複雜到由多個運算子和運算元組成。

4-1-1 關於運算式

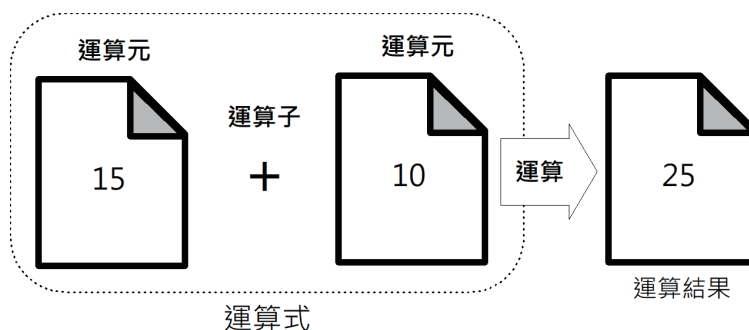
到目前為止，我們已經撰寫過多個 C++ 程式，在說明運算式之前，讓我們先回到程式 (program) 本身，看一看程式到底在作什麼事？在第 2 章是使用標準輸出 cout 輸出執行結果；第 3 章使用標準輸入 cin 數取得輸入資料，事實上，幾乎所有程式都可以簡化成三種基本元素，如下所示：

- ▶ 取得輸入資料。
- ▶ 處理輸入資料。
- ▶ 產生輸出結果。



當然有些程式可能沒有輸入元素 (直接使用常數值，或指定變數值來取代輸入值)，只有輸出元素的執行結果，但是，對於任何有功能的程式，一定少不了處理元素，我們需要使用本章的運算式，第 5~6 章的條件判斷和迴圈來處理輸入資料，以便產生所需的執行結果。

「運算式」(expressions) 是由一序列「運算子」(operators) 和「運算元」(operands) 組成，可以在程式中執行所需的運算任務 (即處理資料)，如下圖所示：

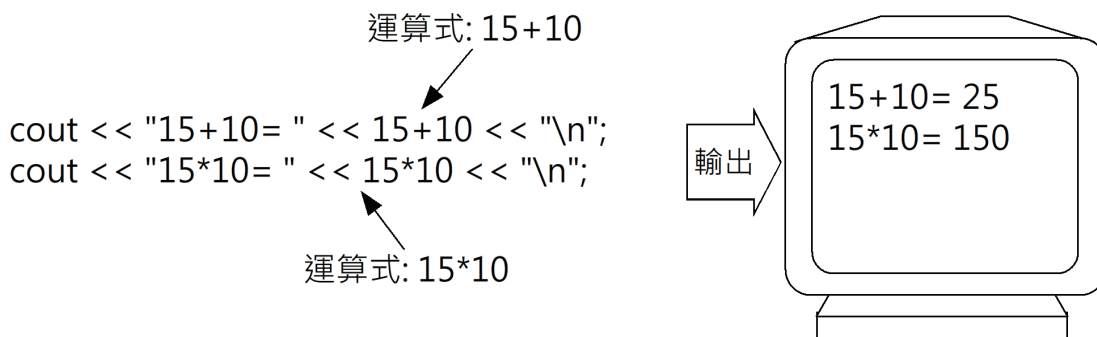


上述圖例的運算式是「15+10」，「+」加號是運算子；15和10是運算元，在執行運算後，可以得到運算結果25，其說明如下所示：

- ▶ 運算子：執行運算處理的加、減、乘和除等運算符號。
- ▶ 運算元：執行運算的對象，可以是常數值、變數或其他運算式。

4-1-2 輸出運算式的運算結果

在第2章我們是使用標準輸出cout在電腦螢幕輸出執行結果，同樣的，我們一樣可以輸出運算式的運算結果，如下圖所示：



上述程式碼計算「15+10」和「15*10」運算式的結果後，輸出顯示在電腦螢幕上。

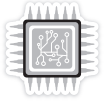


程式語言的乘法是使用「*」符號，而不是手寫「x」符號，因為「x」符號很容易與變數名稱混淆，因為當運算式有x時，我們會將它視為變數；而不是乘法運算子。

ch4-1-2.cpp：輸出運算式的運算結果

```

01: /* 輸出運算式的運算結果 */
02: #include <iostream>
03: using namespace std;
04:
05: int main()
06: {
  
```



```
07: // 計算和輸出 15+10 運算式的值
08: cout << "15+10 = " << 15+10 << "\n";
09: // 計算和輸出 15*10 運算式的值
10: cout << "15*10 = " << 15*10 << "\n";
11:
12: return 0;
13: }
```

ch4-1-2.cpp的執行結果

```
15+10 = 25
15*10 = 150
```

4-1-3 執行不同種類運算元的運算

第4-1-1節說明過運算元可以是常數值或變數，在第4-1-2節運算式的2個運算元都是常數值，除此之外，我們還有2種組合，即2個運算元都是變數，和1個運算元是變數；1個是常數值。

◎ 2個運算元都是變數

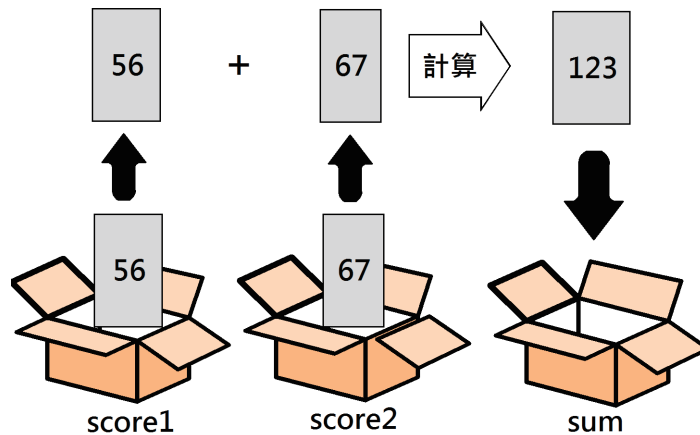
在C++加法運算式的2個運算元可以是2個變數，例如：計算分數的總和，如下所示：

```
int score1 = 56;
int score2 = 67;

int sum = score1 + score2; // 加法運算式
```

上述運算式「score1+score2」的2個運算元都是變數，「sum = score1+score2」運算式的意義是：

「取出變數score1儲存的值56，和取出變數score2儲存的值67後，將2個常數值相加56+67後，再將運算結果123存入變數sum。」



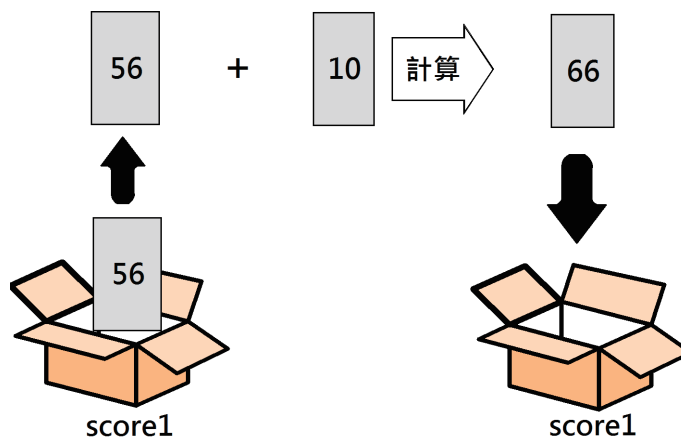
◎ 1個運算元是變數；1個運算元是常數值

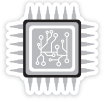
C++加法運算式的2個運算元也可以其中一個是變數；另一個是常數值，例如：調整變數 `score1` 的分數，將它加10分，如下所示：

```
int score1 = 56;  
  
score1 = score1 + 10; // 加法運算式
```

上述運算式「`score1+10`」的第1個運算元是變數；第2個是常數值，「`score1 = score1+10`」運算式的意義是：

「取出變數 `score1` 儲存的值56，加上常數值10後，再將運算結果 $56+10=66$ 存入變數 `score1`。」





請注意！從「score1 = score1 + 10」運算式就可以明顯看出「=」等號不是相等，而是用來指定或指派左邊變數的值，不要弄錯成數學的等於，因為從運算式可以看出，score1 根本不可能等於 score1+10。

ch4-1-3.cpp：執行不同種類運算元的運算

```
01: /* 執行不同種類運算元的運算 */
02: #include <iostream>
03: using namespace std;
04:
05: int main()
06: {
07:     int score1 = 56;        // 第1個運算元
08:     int score2 = 67;        // 第2個運算元
09:     int sum = score1 + score2; // 計算2個變數相加
10:
11:     // 顯示 score1+score2 運算式的運算結果
12:     cout << "變數 score1 值 = " << score1 << "\n";
13:     cout << "變數 score2 值 = " << score2 << "\n";
14:     cout << "score1+score2 = " << sum << "\n";
15:
16:     score1 = score1 + 10;    // 計算變數加常數值
17:     // 顯示 score1+10 運算式的運算結果
18:     cout << "變數 score1 加 10 分 = " << score1 << "\n";
19:
20:     return 0;
21: }
```

ch4-1-3.cpp的執行結果

```
變數 score1 = 56
變數 score2 = 67
score1+score2 = 123
變數 score1 加 10 分 = 66
```



本節運算式的常數值 10 和變數 score1 是運算元，事實上，它們也是一種最簡單的運算式，如下所示：

10

score1

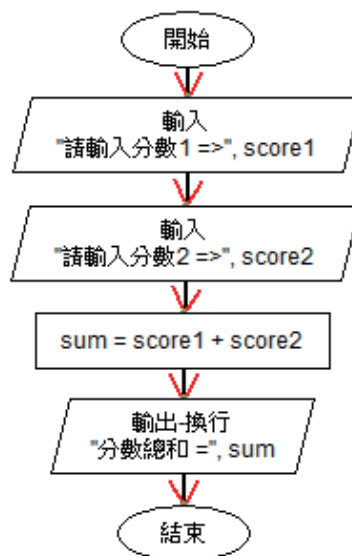
上述常數值 10；變數 score1 是運算式，常數值 10 的運算結果是 10；變數 score1 的運算結果是儲存的常數值。我們可以說，運算式的運算元就是一個運算式，可以簡單到只是一個常數值，或一個變數，也可以是另一個擁有運算子的運算式。

4-1-4 讓使用者輸入值來執行運算

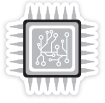
C++ 變數是用來儲存執行時的暫存資料，如果運算式的運算元是變數，我們只需更改變數值，就可以產生不同的運算結果。在這一節的 C++ 程式是修改第 2-3 節範例，可以讓使用者輸入 2 個運算元來計算相加的結果。

◎ 步驟一：觀察流程圖

請啟動 fChart 開啓「CPP\ch04\ch4-1-4.fpp」專案的流程圖，如下圖所示：



請執行流程圖依序輸入 20 和 15，就會顯示相加結果 35，依據流程圖的執行順序，我們可以找出執行步驟，如下所示：



STEP 1 顯示提示文字輸入整數 score1 值（輸入符號）

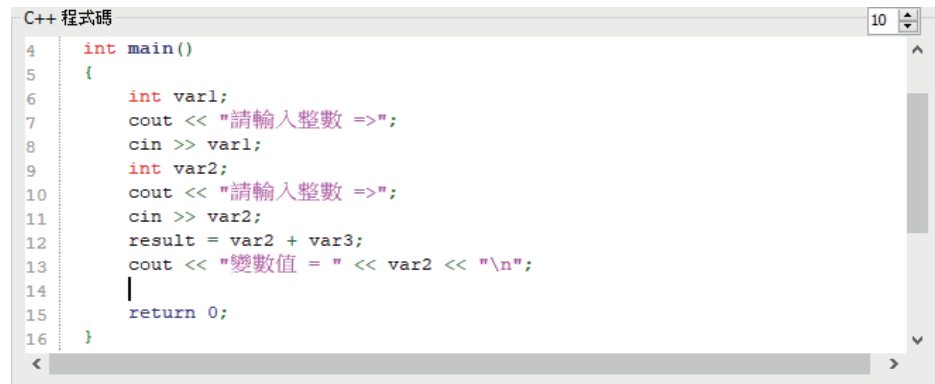
STEP 2 顯示提示文字輸入整數 score2 值（輸入符號）

STEP 3 計算相加結果儲存至 sum（動作符號）

STEP 4 輸出文字內容和 sum 值（輸出符號）

◎ 步驟二：實作程式碼

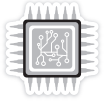
請使用 fChart 程式碼編輯器輸入對應流程圖符號的 C++ 程式碼，首先執行 2 次「輸出/輸入符號>輸入符號>輸入整數值」命令，然後執行「動作符號>算術運算式>加法」命令插入加法運算式，最後加上輸出符號，如下圖所示：



```
C++ 程式碼
4  int main()
5  {
6      int var1;
7      cout << "請輸入整數 =>";
8      cin >> var1;
9      int var2;
10     cout << "請輸入整數 =>";
11     cin >> var2;
12     result = var2 + var3;
13     cout << "變數值 = " << var2 << "\n";
14     |
15     return 0;
16 }
```

在新增變數 sum 宣告後，請修改變數名稱 score1、score2 和訊息文字，就可以建立 C++ 程式 ch4-1-4.cpp，如下所示：

```
06: int score1, sum;
07: cout << "請輸入分數 1 =>";
08: cin >> score1;
09: int score2;
10: cout << "請輸入分數 2 =>";
11: cin >> score2;
12: sum = score1 + score2;
13: cout << "分數總和 = " << sum << "\n";
```

4-5 Arduino 實驗範例：交互閃爍LED燈

在這一節實驗除了原來紅色LED燈，我們準備再新增1個黃色LED燈，然後使用這2個LED燈來切換閃爍的LED燈，程式是使用餘數運算和變數來切換腳位編號。

◎ 電子電路設計

完成本節實驗的電子電路設計需要使用到的電子元件，如下所示：

紅色LED燈 x 1

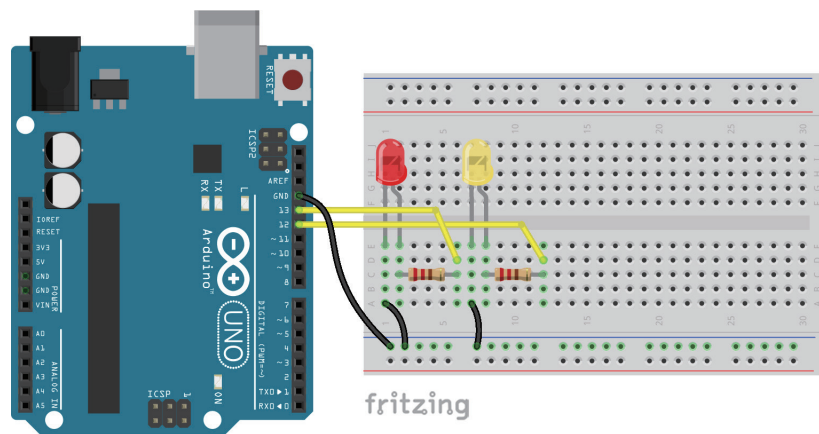
黃色LED燈 x 1

220Ω 電阻 x 2

麵包板 x 1

麵包板跳線 x 5

請依據下圖連接建立電子電路後，紅色LED燈連接在腳位13，黃色連接在腳位12，就完成本節實驗的電子電路設計，如下圖所示：



Arduino 程式：ch4-5.ino

程式使用變數和餘數運算來切換點亮的LED燈，其執行結果可以看到首先閃爍腳位13的紅色LED燈，然後是腳位12的黃色LED燈，接著是13；12，不停切換閃爍2個LED燈，如下所示：

```
01: int ledPin;
02: long count;
03:
04: void setup()
05: {
06:   count = 1;
07:   pinMode(12, OUTPUT);
08:   pinMode(13, OUTPUT);
09: }
10:
11: void loop()
12: {
13:   ledPin = 12 + (count % 2);
14:   digitalWrite(ledPin, HIGH);
15:   delay(1000); // 暫停 1 秒
16:   digitalWrite(ledPin, LOW);
17:   delay(1000); // 暫停 1 秒
18:   count++;
19: }
```

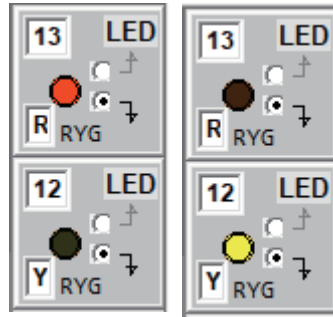
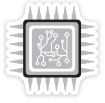
上述第1~2行是2個變數宣告，如下所示：

```
int ledPin;
long count;
```

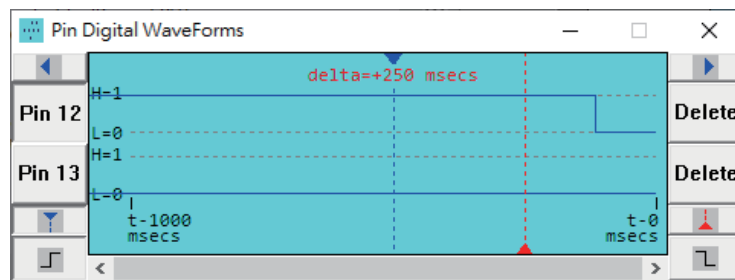
上述ledPin變數儲存腳位編號，值可能是12或13，變數count是計數器，在第7~8行指定腳位12和13的模式都是OUTPUT，第13行使用餘數運算指定此次循環的腳位編號，如下所示：

```
ledPin = 12 + (count % 2);
```

上述程式碼使用餘數運算，值分別是0或1，再加上12後，就是腳位編號12或13，最後在第18行遞增計數變數count的值。當在UnoAduSim模擬器執行時，可以看到右上角腳位13和12的紅色和黃色LED燈在交互閃爍，如下圖所示：



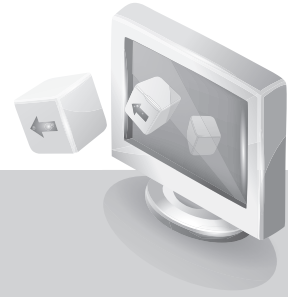
腳位 12 和 13 的數位波型，如下圖所示：



上述波型當腳位 12 是 H=1 時，腳位 13 是 L=0；反之，腳位 13 是 H=1 時，腳位 12 是 L=0。



自我評量



⇒ 選擇題

- () 1. 請問運算式「 $15 + 10$ 」中，哪一個是運算子？
(A)「+」 (B)「15」 (C)「 $0+1$ 」 (D)「10」。
- () 2. 請問下列哪一個是C++ 語言乘法運算子的符號？
(A)「x」 (B)「*」 (C)「X」 (D)「\$」。
- () 3. 請問哪一種是C++ 加法運算式中2個運算元的可能組合？
(A)變數 + 變數 (B)變數 + 常數值 (C)常數值 + 常數值 (D)以上皆是。
- () 4. 請問哪一個並不是C++ 語言的運算子種類？
(A)單元運算子 (B)二元運算子 (C)三元運算子 (D)四元運算子。
- () 5. 請問下列哪一個並不是C++ 語言的算術運算子？
(A)「==」 (B)「-」 (C)「*」 (D)「+」。
- () 6. 請問執行C++ 程式碼片段： $x = 11; y = x++;$ 後，變數y值是什麼？
(A)9 (B)10 (C)11 (D)12。
- () 7. 請問哪一個C++ 語言的運算子優先順序最高？
(A)「++」 (B)「&&」 (C)「*」 (D)「+」。
- () 8. 請問哪一個C++ 語言的運算子是使用右左結合？
(A)「++」 (B)「&&」 (C)「%」 (D)「>=」。
- () 9. 請問我們可以使用下列哪一種語法，將變數a轉換成浮點數型態？
(A)(double)[a] (B)<double>a (C)(double)a (D)[double](a)。
- () 10. 請問下列哪一個是Arduino 程式的副檔名？
(A).c (B).cpp (C).ino (D).ardu。

⇒ 填充與問答題

1. 事實上，幾乎所有程式都可以簡化成三種基本元素：_____、_____和_____。
2. 運算式 (expressions) 是由一序列_____ (operators) 和_____ (operands) 組成。
3. $x = x - y$ 運算式可簡寫成_____， $x = x * y$ 可簡寫成_____。



自我評量



4. 請舉例說明什麼是運算子的優先順序和結合？
5. 請問什麼是資料型態轉換？

⇒ 實作題

1. 請寫出下列C++ 運算式的值，如下所示：

- (1) $1 * 2 + 4$
- (2) $7 / 5$
- (3) $10 \% 3 * 2 * (2 + 5)$
- (4) $1 + 2 * 3$
- (5) $(1 + 2) * 3$
- (6) $16 + 7 * 6 + 9$
- (7) $(13 - 6) / 7 + 8$
- (8) $12 - 4 \% 6 / 4$

2. 請寫出下列C++ 程式碼片段的執行結果，如下所示：

```
int i = 1;
i *= 5;
i += 2;
cout << "i = " << i << "\n";
```

3. 請寫出下列C++ 程式碼片段的執行結果，如下所示：

```
int x = 9;
cout << "x = " << ++x << "\n";
cout << "x = " << x-- << "\n";
```