

## Chapter 1： C/C++程式基礎

### 1.1 C/C++程式基本架構

#### 1.1.1 註解(Comments)

1. 單行註解：`// 註解`
2. 多行註解：`/* 註解一  
註解二 */`

#### 1.1.2 主程式

標題檔

```
void main()  
{  
    程式內容;  
}
```

程式之執行是由主程式首行開始執行，至最後一行執行完畢為止。

程式內容主要包含有：

1. 變數(Variable)與函式(Function)之宣告。
2. 程式命令、函式呼叫。

程式內容的每一段程式命令，均必須要以 `;` 做結尾。

#### 1.1.3 標題檔(Header File)—函式庫之引用

```
#include <standard.h>    //標準標題檔之引用  
#include "myfile.h"     //自訂標題檔之引用
```

欲使用標準的或自訂的函式庫中之函式，需於主程式前將包含欲使用函式的標題檔，使用前處理 `#include` 指令引用進來。

例：`iostream.h fstream.h stdio.h math.h stdlib.h string.h time.h`

程式範例：`cpp_ex1.cpp`

## 1.2 輸出/輸入(I/O)

### 1.2.1 標準輸出/輸入(螢幕/鍵盤)

**C Style:** 使用 `printf & scanf` 定義於 `stdio.h`

例：`printf("CT3407 has %2d students. \n", num);`

**C++ Style:** 使用 `cout & cin` 定義於 `iostream.h`

**cin**：自鍵盤輸入資料

格式：`cin >> 變數 1 [ >> 變數 2 ..... ];`

例：`cin >> x >> y >> z;`

**cout**：將資料輸出置螢幕

格式：`cout << 資料 1 [ << 資料 2 ..... ];`

例：`cout << "This is " << x << y + z << endl;`

註：資料可以是字串(置於"內)、變數(顯示的為變數之值)、運算式(顯示的為運算結果)。

換行指令：

`endl` 或 `"\n"`

程式範例：[cpp\\_ex2.cpp](#)

### 1.2.2 檔案輸入/輸出

使用 `<fstream>` 中的 `ifstream` 與 `ofstream` 指令。

格式：

`#include <fstream>`

`using namespace std;`

.....

`ifstream 變數名 = ifstream("輸入檔檔名");`

`ofstream 變數名 = ofstream("輸出檔檔名");`

例：

```
ifstream cin_from_file = ifstream("data.txt");
ofstream cout_to_file = ofstream("result.txt");
```

程式範例：[cpp\\_ex3.cpp](#)

## 1.3 運算子(Operator)

### 1.3.1 設定運算子(Assignment Operator) “=”：

格式：變數 = 數值/變數/運算式；

說明：把“=”右邊的數值/變數值/運算式之結果值，設定給“=”左邊的變數。

例：x = 30;

x = x + 20; // C/C++中，= 非“等於”，而是“設定”

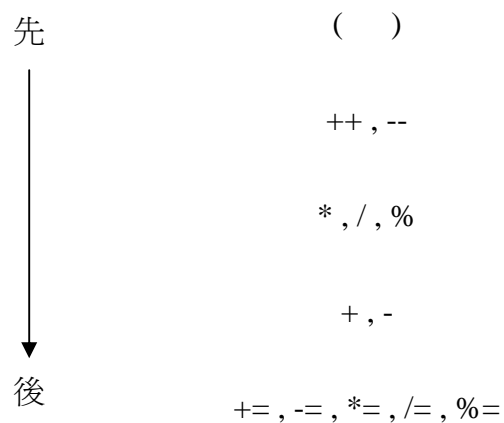
程式範例：[cpp\\_ex4.cpp](#) (變數型態轉換)

### 1.3.2 算術運算子(Arithmetic Operator)：

運算子	例	作用
+	a + b	a 加 b
-	a - b	a 減 b
*	a * b	a 乘 b
/	a / b	a 除以 b
%	a % b	a 除以 b 之餘數
++	i++, ++i	對 i 的值加 1 (i = i + 1)
--	i--, --i	對 i 的值減 1 (i = i - 1)
+=	a += b	a 加 b，結果存入 a (a = a + b)
-=	a -= b	a 減 b，結果存入 a (a = a - b)
*=	a *= b	a 乘 b，結果存入 a (a = a * b)
/=	a /= b	a 除以 b，結果存入 a (a = a / b)
%=	a %= b	a 除以 b 之餘數，存入 a (a = a % b)

程式範例：`cpp_ex5.cpp` (++ , --)

算術運算子之執行優先順序：



### 1.3.3 關係運算子(Relational Operator)：

運算子	例	作用
==	a == b	判斷 a 是否等於 b
!=	a != b	判斷 a 是否不等於 b
>	a > b	判斷 a 是否大於 b
<	a < b	判斷 a 是否小於 b
>=	a >= b	判斷 a 是否大於或等於 b
<=	a <= b	判斷 a 是否小於或等於 b

判斷結果：

1. 是，或成立：傳回結果為 `true` (1)
2. 否，或不成立：傳回結果為 `false` (0)

程式範例：`cpp_ex6.cpp`

### 1.3.4 邏輯運算子(Logical Operator)：

運算子	意義	說明
&&	且(and)	當運算子左右均為 true，才傳回 true
	或(or)	當運算子左右其一為 true，即傳回 true
!	非(not)	將 true 改為 false，將 false 改為 true

邏輯運算子通常用於連結關係運算式(結果非 true 即 false)或布林值(可進一步廣泛用到整數)，其運算結果如下：

	關係式 A	關係式 B	A && B	A    B	!A
Case 1	true	true	true	true	false
Case 2	true	false	false	true	false
Case 3	false	true	false	true	true
Case 4	false	false	false	false	true

由關係運算子或邏輯運算子所組成的運算式，成為布林(Boolean)運算式，其結果必為 true 或 false，其用於程式中的決策判斷。

程式範例：[cpp\\_ex6.cpp](#)

### 1.3.5 運算子之執行優先順序

