博碩書號:PG20266

C++的基本知識

第2章

博碩文化出版發行

2-1 輸出結果到電腦螢幕

- 輸入新的程式碼
- 輸出結果到電腦螢幕
- 輸出到電腦螢幕的方式

輸入新的程式碼

- 接著將以下列程式為範例，來說明程式的每個部份：

Sample1.cpp

```
//輸出文字到螢幕的範例程式碼
#include <iostream>
using namespace std;

int main()
{
    cout << "歡迎來到C++！\n";
    cout << "開始使用C++吧！\n";
    return 0;
}
```

註解內容

用來輸出至畫面

main()函數的開始

從這一行開始執行

接下來執行這一行

main()函數的結束

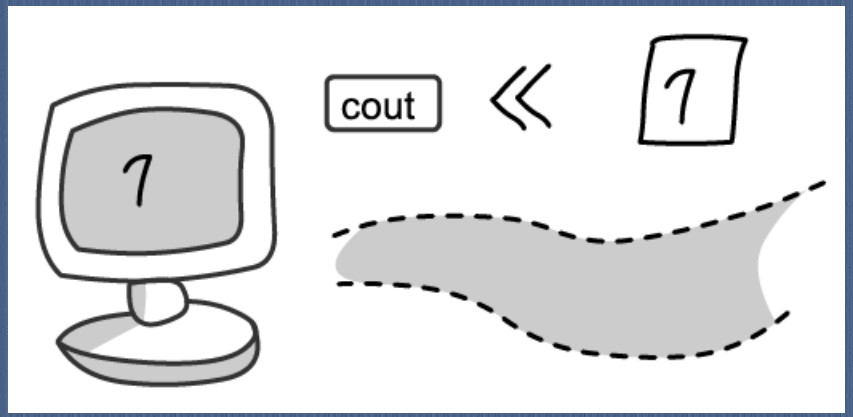
輸出結果到電腦螢幕

```
#include <iostream>
using namespace std;
int main()
{
    cout << 想要輸出的文字或數值 ;
    return 0;
}
```

上面的範例是要輸出文字或數值到畫面時的基本程式語法。

輸出到電腦螢幕的方式

- 所謂的「標準輸出」是指「目前連接在電腦上的螢幕」。
- 「cout」就是標準輸出（standard output）。
- 「<<」是指將文字或其他資料列印（＝顯示）到電腦螢幕的意思。
- cout...這一行程式碼可以「把文字或其他資料輸出到電腦螢幕並顯示出來」。



- 換行的程式範例：

```
#include <iostream>
using namespace std;
int main()
{
   cout << 1 << 2 << 3 << '\n' << 4 << 5 << '\n';
   return 0;
}
```

- 執行畫面：

```
123
45
```

- 「'\n'」就是執行換行的意思。

2-2 程式碼的內容

- 檢視程式的執行流程
- main()函數
- 以敘述句（statement）為單位執行程式
- 讓程式碼更簡易、易讀
- 註解（comment）
- 引入檔案

檢視程式碼與執行流程

```
//輸出文字到螢幕的範例程式碼
#include <iostream>
using namespace std;

int main()
{
    cout << "歡迎來到C++！\n";
    cout << "開始使用C++吧！\n";
    return 0;
}
```

註解內容

為了使用**cout**而做的記述

main()函數的開始

從這一行開始執行

接下來執行這一行

main()函數的結束

瞭解MAIN()函數

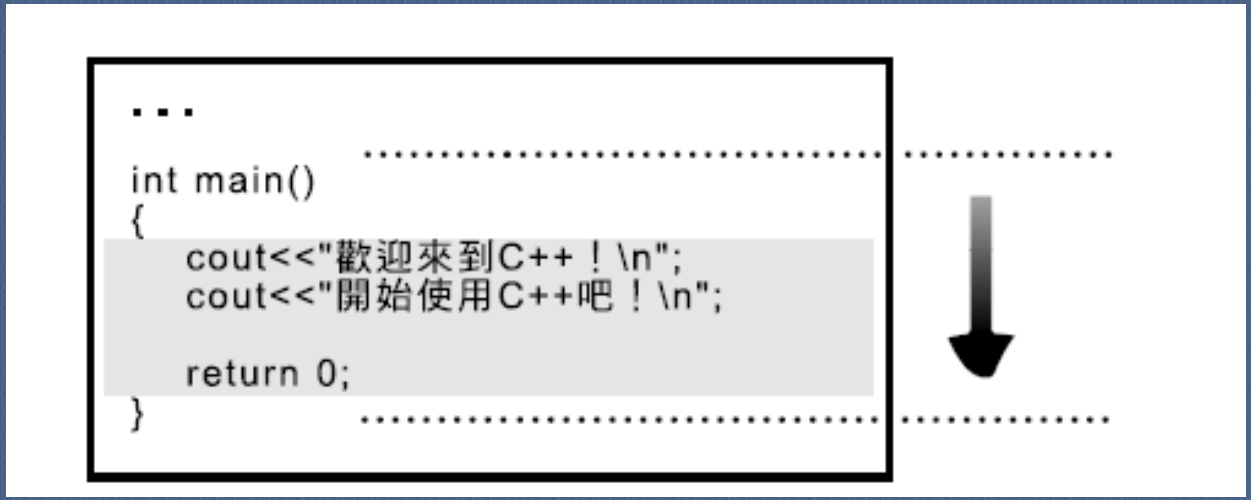
- main()是程式的主體
  - 由2個大括號「{}」所涵蓋的所有程式碼，就稱為程式區塊（block），它是程式裡最重要的主體部分，又稱為main()函數（main function）。
  - 範例：

```
int main()
{
   …
   return 0;
}
```

main()函數 (程式的主體)

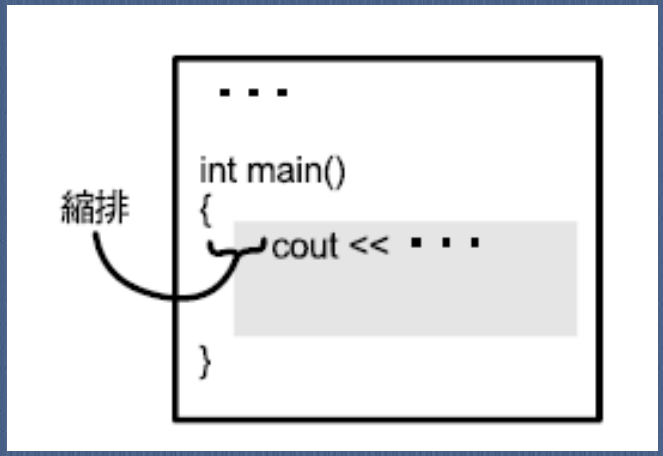
以敘述句為單位執行程式

- C++語言會將每一個要執行的工作，劃分為一行一行的小單元，每一行就是一個「敘述句（statement）」，敘述句的最後要加上「；（分號）」。
- 在執行statement時，原則上會根據程式的先後順序，從最前面的statement開始一行接著一行執行。



讓程式碼更簡易、易讀

- 不論是在statement或程式區塊中，若有必要的話都可以換行。
- 在C＋＋中，只要地點正確且不會破壞程式的完整性，其實很多地方都可以插入空格或換行。
- 將程式區塊內的文字適度往內**縮排**，可以使程式層次分明、簡潔易懂。適當的程式縮排或換行是讓程式清楚易懂的不二法門。



註解（COMMENT）

- // 後面的文字與程式的執行結果無關，您可以在// 後面輸入程式註解或是任何您想要輸入的文字。
  - 例如：

    //輸出文字到螢幕的範例程式碼

- 使用註解來輔助說明，能夠讓程式碼的內容更易懂。
- 另一種程式註解的方法：
  - 例如

    /* 將文字

    輸出到螢幕上 */

- 所有包圍在/**/中的文字都算是註解的一部分。

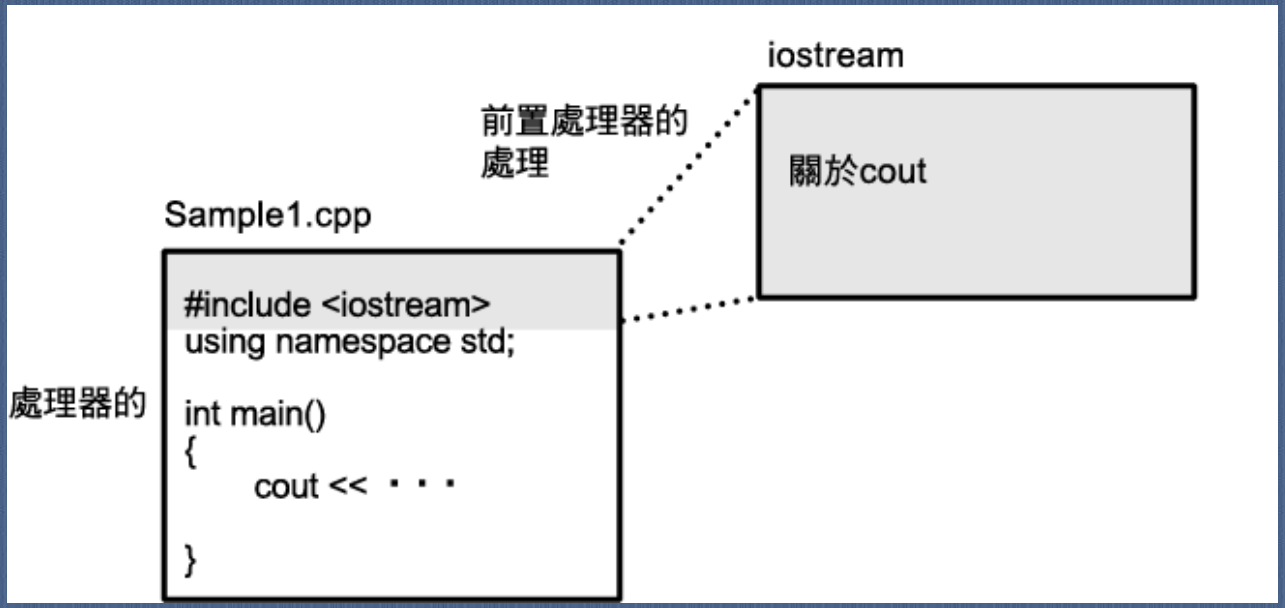
引入檔案

#include <iostream>

- 若要執行輸出到螢幕的指令，就要在編譯之前先行引入iostream這個檔案，它是用來規定輸出至螢幕之功能的檔案，而載入這個檔案的作業就稱為「引入(include)」。
- 在C++裡，像這種要事先引入的檔案就稱為標頭檔（header file）。
- 帶有#的這一行會透過前置處理器處理，並在編譯其他程式碼之前讀入，它只能寫成一行且最後不能加分號。

using namespace std;

- 「cout」的正式寫法原本是std.cout ，但只要在程式碼的前頭先註明「using namespace std;」，就可以偷懶只寫「cout」了。



2-3 文字和數值

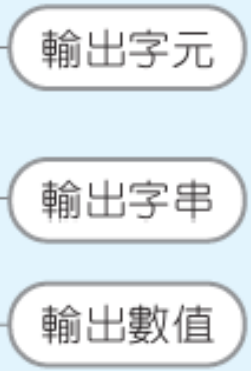
- 何謂字面量(literal)？
- 字元字面量
- 跳脫字元
- 字元碼
- 字串字面量
- 數值常數

何謂字面量(LITERAL)？

- 字面量(或稱為常值)是一種通稱，泛指字元、字串、數字等各種型態的資料。
- 如果要仔細劃分字面量的話，又可將它分為下列這4種：
  - 字元字面量
  - 字串字面量
  - 整數字面量（整數常數）、浮點數字面量（浮點數常數）
  - 邏輯字面量

Sample3.cpp ▶ 輸出各種資料

```
#include <iostream>
using namespace std;

int main()
{
    cout << 'A' << '\n';
    cout << "歡迎來到C++！\n";
    cout << 123  << '\n';

    return 0;
}
```

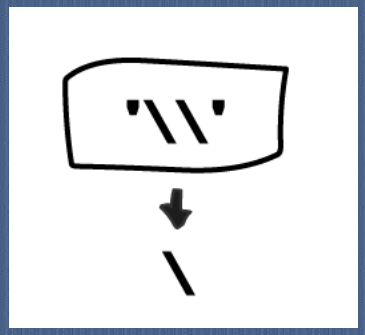


字元字面量

- 文字字面量可分成以下2種：
  - 字元（單一字母）
  - 字串（數個字母）
- 字元就是程式碼中以兩個單引號(‘ ’)所圍住的部分，如下所示。
  - 'H'
  - 'e'
- 通常只有英文字母與數字算是字元，單一中文字的處理可能會出現不同的結果。

跳脫字元

- 有一些單一字元無法表示的特殊文字，這些特殊文字的前面必須加上「\」符號，並與原來的字元組合後才能產生新的意義。這樣的特殊文字通稱為跳脫字元（escape sequence），右邊為其列表：



爲了顯示出特殊字元，就必須使用跳脫字元。

跳脫字元	說明
\a	警告音
\b	BackSpace
\t	水平tab
\v	垂直tab
\n	換行
\f	換頁
\r	return
\\	\
\'	'
\"	"
\?	?
\ooo	8進制ooo字元碼（o表0～7的數字）
\xhh	16進制hh字元碼（h表0～9的數字與A～F的英文字母）

Sample4.cpp ▶ 特殊文字的輸出

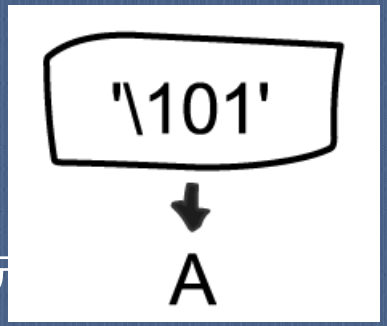
```
#include <iostream>
using namespace std;

int main()
{
    cout << "顯示出反斜線：" << '\\' << '\n';
    cout << "顯示出單引號：" << '\'' << '\n';

    return 0;
}
```

此處使用跳脫字元

字元碼

- 事實上在電腦內部處理文字時，也是將字元轉換成「特殊的數字」再加以處理，也因此每個字元都有其相對應的數字，這就叫做字元碼（character code）。字元碼的種類很多，例如有BIG-5、Unicode（萬國碼）等字元碼系統，究竟要使用哪一種字元碼，應視使用環境加以選擇。

也可以指定字元碼來顯示字元



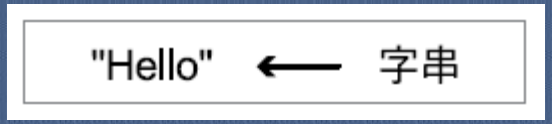
Sample5.cpp ▶ 設定字元碼並輸出結果

```
#include <iostream>
using namespace std;

int main()
{
   cout << "8 進位數 101 的字元是 " << '\101' << " ◦ \n" ;
   cout << "16 進位數 0061 的字元是 " << '\x61' << " ◦ \n" ;

   return 0;
}
```

在此指定字元碼

字串字面量

- 相較於「字元」是由單一字母所構成，字串字面量（string literal）則是由2個或2個以上的字母所組合而成，因此兩者的表示方法也不相同。字元是使用「''」、而字串則是使用「""」將其內容包圍起來：
  - "Hello"
  - "Goodbye"

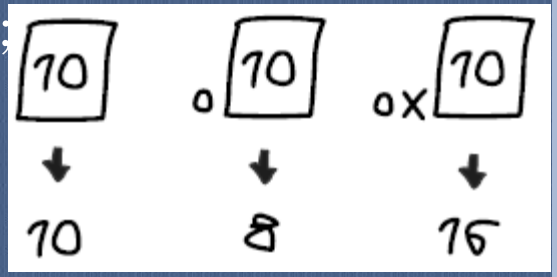


數值字面量

- C語言也可以用來處理數字，包含下列這兩種：
  - 整數字面量（integer literal）
    例如1、3、100等
  - 浮點數字面量（floating literal）
    例如2.1、3.14、5.0等

- 我們除了可以用一般的方式來標示數字之外，還有8進制、16進制等方式都可以標示數字：
  - 8進制……在數字的最前面加上0
  - 16進制……在數字的最前面加上0x

```
cout << "10進位數的10是" << 10 << "。\n";
cout << "8進位數的10是" << 010 << "。\n";
cout << "16進位數的10是" << 0x10 << "。\n";
cout << "16進位數的F是" << 0xF << "。\n";
```



綜合整理

- 本章學習過的內容與重點
  - main()函數是C++程式的主體。
  - 程式的敘述句（statement）是程式處理的最小單位。
  - 程式區塊（block）是程式處理的較大單位。
  - 程式中寫在//後面的文字就稱為註解，善用註解可以提高程式的可讀性。
  - 字面量是一種通稱，它可以是字元、字串或數值。
  - 字元的外圍要用' '框住。
  - 某些特殊符號必須透過跳脫字元才能顯示。
  - 字串的外圍要用" "框住。
  - 整數常數（integer literal）除了可以使用10進制來表示外，還可以使用8進制、16進制表示。