chapter

2

常數變數與資料

本章重點▶

2.1　C++ 程式結構

2.2　常數與變數

2.3　C++ 資料型態

2.4　習題

碁峰
www.gotop.com.tw

2.1 C++ 程式結構

一般而言，一個複雜的 C++ 大程式是由許多不同檔案的小模組所組成，而一個簡單的 C++ 小程式可儲存於單一檔案中。然而一個簡單的 C++ 小程式可概分為七部分：程式註解區、前置處理區、公用變數區、程式起始區、區域變數區、程式敘述區、與程式結束區等。

（一）程式註解是給人看的，它不影響編譯與執行的結果，它可出現在任何需要說明的地方，包括出現在指令結束之後。（二）前置處理區提供如 #include 敘述載入程式所需的標題檔，標題檔包含程式所需呼叫的函數，因此前置處理區必須置於程式執行之前，也就是 main 函數宣告之前。（三）公用變數區也包含程式所需使用的變數，所以通常置於所有函數之前，以提供程式中所有函數的敘述使用。（四）程式起始區包含 main 函數與函數起始符號（左大括號），main 函數是每個 C++ 程式必要的函數，當執行 C++ 程式時，作業系統會呼叫 main 函數。（五）區域變數區通常置於該函數的前端，以供該函數的所有敘述使用，但變數若置於前置處理區，則為公用變數，如此所有函數的敘述皆可使用。（六）程式敘述區則一定包含於 main 函數或其他函數中。（七）程式結束區包含 system 與 return 指令與函數結束符號（右大括號），system("PAUSE")是令系統暫停執行直到按任意鍵才繼續，return EXIT_SUCCESS 則傳回成功結束代碼給系統。表 2.1 列出圖 2.1 程式中所使用的符號與符號的用途說明。

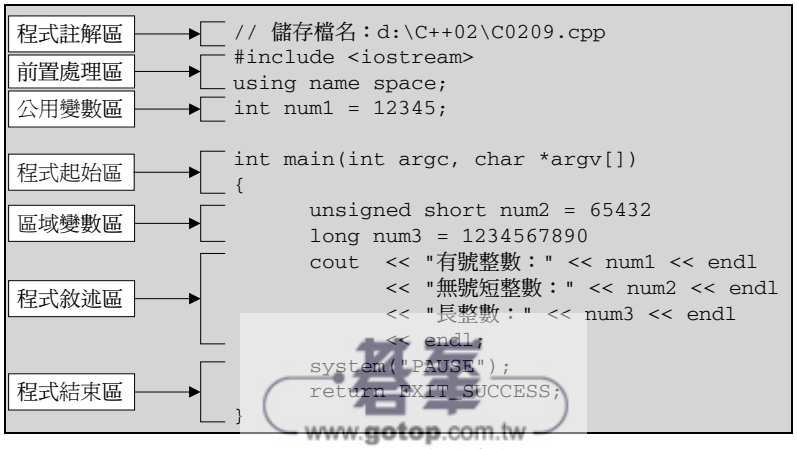


圖 2.1 C++ 程式架構

表 2.1 C++的特殊符號

字元	名稱	說明
//	雙斜線	為註解的起始符號，如 //檔案名稱：d:\C++02\C0209.cpp
#	井號	為前置處理指令的起始符號，如 #include
< >	角括號	當使用 #include 指令時，用來包含要引入的標題檔名
( )	小括號	用於宣告函數，小括號內包含函數參數，如 int main(void)
{ }	大括號	用於包含一個或多個敘述，如 { cout << endl; }
" "	雙引號	用於包含一個字元串列，如 "Hello World!"
;	分號	為敘述的結束符號，如 return 0;

2.1.1 程式註解 //

// 註解
/* 註解 */

- **註解**提供原始程式的輔助說明，它雖然屬於程式的一部分，但編譯器將忽略註解。
- // 是 C++ 型態的註解起始符號，所以雙斜線後的文字皆視為說明文字。
- /* */ 是 C 語言型態的註解，/* */ 範圍內的文字也是說明文字。

下面範例的第一式是 C++ 型態的新型註解，第二是則是 C 型態的舊型註解。

```
//儲存檔名：d:\C++02\C0201.cpp
/* 宣告整數變數練習 */
```

碁峰 www.gotop.com.tw

2.1.2 插入標題檔 #include

```
#include <標題檔名>                    // 第一式
#include "標題檔名"                    // 第二式
```

- **#include** 是將標題檔案插入到程式內。例如因 cout 與 cin 函數不包含於 C++ 語言核心的函數，而是屬於輸入/輸出串列資料庫的一個函數，cout 定義在 iostream 新型標題檔中的函數（舊型標題檔為 iostream.h），所以使用 cout 或 cin 必須先插入 iostream（或 iostream.h）到程式內。

```
#include <iostream.h>
```

- **#include** 是前置處理指令，不是 C++ 的敘述。所謂前置處理指令就是在 C++ 編譯敘述之前，將程式中所參考的函數或類別插入到 #include 指令之後，如此當 C++ 編譯時才不會產生錯誤。

```
#include <iostream>                          //插入 iostream 檔
using namespace std;                         //宣告使用標準函數庫
int main(int argc, char *argv[])
{
   cout << "Hello World!";                   //正確顯示字串
   return EXIT_SUCCESS;                      //傳回成功結束碼給系統
}
```

- 第一式的角括號 <> 用來插入 C++ 程式庫標題檔，第二式的雙引號 "" 則用來插入使用者自訂的標題檔。

```
#include "user.h"                            //插入使用者標題檔
```

表 2.2 列出舊型標題檔與檔案說明，表 2.3 則列出新型標題檔與檔案說明，新型標題檔包含於標準函數庫中，所以使用新型標題檔必須必須加上 using namespace std; 指令，宣告使用標準函數庫。本書全部範例皆採用新型標題檔。

碁峰 www.gotop.com.tw

2.1.3 main() 函數

```
傳回型態 main(參數)
{
.
        return 傳回值;
}
```

- C++ 程式是由一個或多個函數所組成，每個 C++ 程式在 MS-DOS 環境下都包含 main() 函數。
- **傳回型態**表示函數傳回值的資料型態，若傳回型態為 void 表示該函數不傳回任何值給 MS-DOS 系統。
- **參數**則是由 MS-DOS 傳遞給 main 函數的值，可以傳遞數值、變數、指標、陣列等參數。若省略參數或參數為 void 表示該函數不接收任何參數。
- main 的敘述區包含於二個大括號（{ }）中，左大括號表示 main 函數敘述區的起始點，右大括號表示 main 函數敘述區的結束點。
- **return** 是返回作業系統敘述，傳回值則是要傳回給作業系統的數值，若傳回型態為 void 表示不須傳回任何值給作業系統所以可以省略 return 敘述。對於詳細的函數定義、呼叫與返回請參閱 6.1 節。

下面範例的 main 前面的 void 宣告不傳回任何值給作業系統，而小括號內的 void 則宣告在作業系統下執行此程式也不須傳遞任何參數給 main 函數。

```
void main(void)                          //無參數呼叫也不傳回任何值
```

下面範例的 main 前面的 void 宣告不傳回任何值給作業系統，而小括號內省略任何參數則表示在作業系統下執行此程式也不須傳遞任何參數給 main 函數。

碁峰 www.gotop.com.tw

```
void main()                              //省略 void 仍為無參數呼叫
```

下面範例的 main 前面的 int 宣告會傳回整數值給作業系統，而小括號內的 void 則宣告在作業系統下執行此程式也不須傳遞任何參數給 main 函數。

```
int main(void)                                  //無參數呼叫但須傳回整數
```

下面範例的 main 前面的 int 宣告會傳回整數值給作業系統，而小括號內省略任何參數則表示在作業系統下執行此程式也不須傳遞任何參數給 main 函數。

```
int main()                                      //省略 void 仍為無參數呼叫
```

以上範例皆為 ANSI C++ 合法的語法，但是使用 Dev-C++ 建立專案時，會自動產生如下面範例的 main 函數與 return 指令，若修改 Dev-C++ 自動產生的 main 函數格式，則 Dev-C++ 在編譯時可能會產生錯誤訊息，所以本書所有範例皆使用 Dev-C++ 自動產生的 main 函數與 return 指令如下面範例。

```
int main(int argc, char *argv[])
{                                               //main 函數起始點
    //插入敘述區
    system("PAUSE");                            //暫停，等待按任意鍵結束
    return EXIT_SUCCESS;                        //傳回成功結束碼給系統
}                                               //main 函數結束點
```

2.1.4 輸出函數 cout

cout << 變數或字串 1 << 變數或字串 2 << . . . << 變數或字串 n;

- **cout** 是螢幕輸出函數，**<< 為串列輸出（stream output）**運算符號也就是將指定變數中的資料或字串常數資料依序向輸出設備移出，在此是依序輸出到螢幕上。輸出時可利用 endl 函數控制跳行或利用表 2.4 的特殊符號輸出特殊字元與控制碼。

下面範例的第三式是輸出串列包含字串、變數、與控制碼，所以使用三個串列輸出（<<）分別輸出字串、變數、與控制碼。**endl（end of line）**為結束輸出行，因此下一次輸出將跳至下一行起頭。

碁峰 www.gotop.com.tw

```
cout << num1;                          //顯示變數 num1 的值
cout << "ANSI/ISO C++";                //顯示字串 ANSI/ISO C++
cout << "有號整數：" << num1 << endl;    //顯示字串、數值、跳行
```

程式 2-01：cout 練習

```
1.   // 儲存檔名：d:\C++02\C0201.cpp
2.   #include <iostream>
3.   using namespace std;
4.
5.   int main(int argc, char *argv[])
6.   {
7.      cout << "紅葉晚蕭蕭，長亭酒一瓢。";        //輸出字串
8.      system("PAUSE");
9.      return EXIT_SUCCESS;
10.  }
```

程式輸出

```
紅葉晚蕭蕭，長亭酒一瓢。請按任意鍵繼續 . . .
```

程式 2-02：二個 cout 練習

```
1.   //儲存檔名：d:\C++02\C0202.cpp
2.   #include <iostream>
3.   using namespace std;
4.
5.   int main(int argc, char *argv[])
6.   {
7.      cout << "紅葉晚蕭蕭，";                  //輸出字串游標停在最後面
8.      cout << "長亭酒一瓢。";                  //接著輸出第二個字串
9.      system("PAUSE");
10.     return EXIT_SUCCESS;
11.  }
```

程式輸出

```
紅葉晚蕭蕭，長亭酒一瓢。請按任意鍵繼續 . . .
```

程式 2-03：多個 cout 練習

```
1.   //儲存檔名：d:\C++02\C0203.cpp
2.   #include <iostream>
3.   using namespace std;
4.
5.   int main(int argc, char *argv[])
6.   {
7.      cout << "紅葉晚蕭蕭，長亭酒一瓢。";        //輸出字串游標停在最後面
```

碁峰 www.gotop.com.tw

表 2.4 特殊字元表

字元值	字元格式	字元功能
0	\0	空格（null space）
7	\a	響鈴（bell ring）
8	\b	倒退（backspace）
9	\t	移到下一定位點（tab）
10	\n	插入新行（newline）
12	\f	跳至下一頁起點（form feed）
13	\r	跳至同一行起點（carriage return）
34	\”	插入雙引號（double quote）
39	\’	插入單引號（single quote）
92	\\	插入反斜線（back slash）

2.2 常數與變數

變數（variable）代表電腦記憶體中的一個儲存位置，而**常數（constant）**在程式執行中是不可改變的資料項目。

2.2.1 宣告變數

資料型態　變數名稱 1, 變數名稱 2, …;

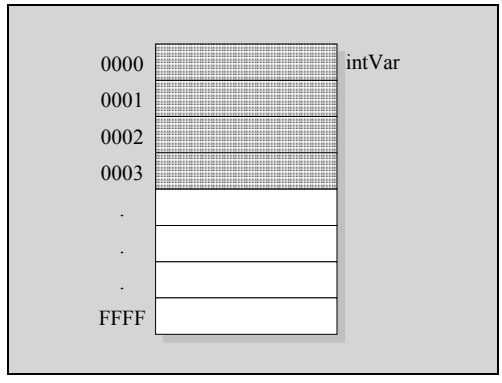
- **宣告變數（variable declaration）**就是告訴編譯器在電腦的記憶體中配置一個指定資料型態的記憶空間，在程式執行中則以此指定的變數名稱來存取該記憶空間。
- **變數名稱（variable’s name）**是英文字母（a-z, A-Z）、數字（0-9）與底線符號（_）的組合，且必須以英文字母為開頭。變數型態表示該變數名稱可以存放數值的型態，如整數（int）、浮點數（float）、字元（char）等等。2.3.1 節至 2.3.4 節將更詳細介紹各型態的資料。

碁峰 www.top.com.tw

- **C++** 是區分大小寫（case sensitive）的程式語言，所以變數 value 與變數 Value 是不同的名稱。

```
int intVar;                                    //宣告整數型態的變數 intVar
```

當 C++ 編譯 int intVar; 敘述時，將在記憶體中配置 4 個 byte 的記憶體空間如下圖，並以 intVar 作為存取該記憶體空間的名稱。



2.2.2 指定資料 =

資料型態　變數名稱 1, 變數名稱 2, ...;
變數名稱 1 = 初值 1;
變數名稱 2 = 初值 2;
...;

- **等號（=）**用來起始與指定變數值。當宣告變數後，則可利用等號（=）給予該變數初值，設定初值後也可任意更改變數值，不過起始值與指定值的型態必須符合宣告的資料型態。
- 使用等號（=）指定變數的等值時，必須視每個對等式為一個敘述，所以等式後面必須加分號（;）。

```
short shortVar;                                //宣告短整數變數 shortVar
shortVar = 5;                                  //shortVar 的初值等於 5
.
shortVar = 10;                                 //改變 shortVar 的值為 10
```

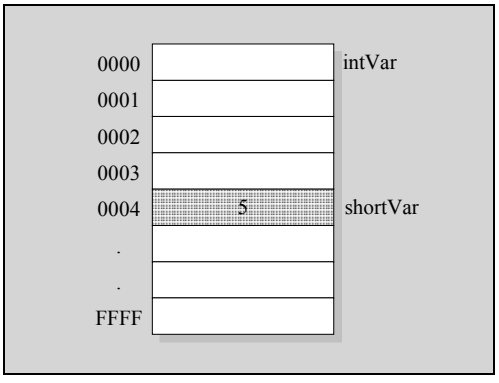
碁峰 www.gotop.com.tw

資料型態　變數名稱 1＝初值, 變數名稱 2＝初值, ...;

- 也可以宣告變數同時指定該變數的初值。此時，整個宣告視為一個敘述，所以每個等式間只須以逗點（,）隔開。

在下面範例中，當 C++ 編譯 short shortVar; 敘述時，將在記憶體中配置 1 個 byte 的記憶體空間並存入數值 5 如下圖，並以 shortVar 作為存取該記憶體空間的名稱。

```
short shortVar = 5;                    //宣告短整數 shortVar = 5
```



程式 2-07：宣告變數、起始與更改變數練習

```
1.   //儲存檔名：d:\C++02\C0207.cpp
2.   #include <iostream>
3.   using namespace std;
4.
5.   int main(int argc, char *argv[])
6.   {
7.      int Var = 5;                              //宣告 Var = 5 (起始值)
8.
9.      cout  << "Var 起始值 = " << Var;          //顯示訊息字串與 Var 值
10.     Var = 10;                                 //改變 Var = 10(變更值)
11.     cout  << "\nVar 變更值 = " << Var << endl; //顯示訊息字串與 Var 值
12.     system("PAUSE");
13.     return EXIT_SUCCESS;
14.  }
```

碁峰 www.gotop.com.tw

程式輸出

```
Var 起始值 = 5
Var 變更值 = 10
請按任意鍵繼續 . . .
```

2.2.3 宣告常數 const

const 資料型態　常數符號 1＝數值 1, 常數符號 2＝數值 2, ...;

- **const** 是指定一個固定值給符號名稱。
- **常數符號**的命名方式與變數名稱相同。
- **資料型態**可以是任何形式的變數型態（如 int、double、float...）。
- **數值**則是與資料型態相符的固定值。
- 宣告常數符號時，必須同時賦予初值，而且符號被宣告為常數後，則該符號的值不能再被改變。
- C++ 編譯宣告常數符號敘述時，並不會配置記憶體空間給該常數符號，只是令該符號等於其指定值。因此當編譯同一程式的其他敘述時，若遇到相同的符號時，則直接以該符號的等值取代。
- 雖然常數符號的值經宣告後即不能再改變，但是常數符號也可用於運算式中。

下面範例宣告浮點常數符號時，最好使用 f 作為等號右邊常數 3.14159 的結尾，表示該常數 3.14159f 為浮點常數值。否則有些 C++ 編譯器會將浮點常數 3.14159 視為倍精度常數值，而編譯時將產生"truncation from 'const double' to 'const float'"的警告訊息。

```
const float fPI = 3.14159f;                    //宣告浮點常數符號 fPI
const double dPI = 3.141592653;                //宣告倍精數常數符號 dPI
```

程式 2-08 第 13 與 18 行的 pow(radius, 2) 是計算 $(radius)^2$ 值，pow 函數的傳回型態為 double，而 float((pow(radius, 2)) 則將 double 型態的值轉

碁峰 www.gotop.com.tw

成 float 型態並存入 area 中。冪次的計算請參閱 3.6.4 節，而資料型態轉換請參閱 3.4.3 節。

程式 2-08：宣告常數符號練習

```
1.   //儲存檔名：d:\C++02\C0208.cpp
2.   #include <iostream>
3.   #include <cmath>
4.   using namespace std;
5.
6.   const float PI = 3.14159f;                  //宣告浮點常數符號 PI
7.
8.   int main(int argc, char *argv[])
9.   {
10.     float area, circu;                       //宣告浮點數 area,circu
11.     float radius = 5;                        //宣告浮點數 radius=5
12.
13.     area = PI * float(pow(radius, 2));       //area=圓面積 1
14.     circu = 2 * PI * radius;                 //circu=圓周長 1
15.     cout << "圓面積 1 = " << area
16.          << "\t 圓周長 1 = " << circu;
17.     radius = 10;                             //改變 radius=10
18.     area = PI * float(pow(radius, 2));       //area=圓面積 2
19.     circu = 2 * PI * radius;                 //circu=圓周長 2
20.     cout << "\n 圓面積 2 = " << area
21.          << "\t 圓周長 2 = " << circu << endl;
22.     system("PAUSE");
23.     return EXIT_SUCCESS;
24.  }
```

程式輸出

```
圓面積 1 = 78.5397        圓周長 1 = 31.4159
圓面積 2 = 314.159        圓周長 2 = 62.8318
請按任意鍵繼續 . . .
```

2.2.4 宣告符號 #define

#define 對等符號 對等資料

- **#define** 指令是早期 C 語言宣告常數符號的方法，雖然 C++ 語言提供 const 來宣告常數符號，但仍有許多程式使用 #define 指令。

碁峰 www.gotop.com.tw

- **對等符號**的命名方式與變數名稱相同。但大多數程式設計師喜歡使用大寫符號來區分該符號是使用 #define 宣告的符號。
- **對等資料**是在程式編譯前，以此對等資料取代程式中所有的對等符號。
- **C++ 前置處理器（preprocessor）**在編譯之前會先掃描程式，因此前置處理指令（preprocessor directives）將優先被處理，例如 #include 會先插入指定的標題檔， #define 則先置換程式中相同的符號。
- 對設計師而言，使用 const 與#define 宣告常數符號是相同的，但 C++ 編譯器處理 const 與 #define 的過程則不同。
- #define 是前置處理指令，所以不使用分號結尾。

下面範例在編譯之前，前置處理器將以 3.14159 取代程式中所有的符號 PI，然後才開始編譯。所以 circumference = 2 * PI * radius; 將被取代為 circumference = 2 * 3.14159 * radius; 後才開始編譯。

```
#define PI 3.14159
int main(int argc, char *argv[])
{
   float circumference, radius = 10;
   circumference = 2 * PI * radius;            //2 * 3.14159 * radius
   return EXIT_SUCCESS;                        //傳回成功結束碼給系統
}
```

程式 2-09 第 13 與 18 行的 pow(radius, 2) 是計算 $(radius)^2$ 值，pow 函數的傳回型態為 double，而 float((pow(radius, 2)) 則將 double 型態的值轉成 float 型態並存入 area 中。冪次的計算請參閱 3.6.4 節，而資料型態轉換請參閱 3.4.3 節。

程式 2-09：宣告常數練習

```
1.   //儲存檔名：d:\C++02\C0209.cpp
2.   #include <iostream>
3.   #include <cmath>
4.   using namespace std;
5.
6.   #define PI 3.14159f                         //宣告符號 PI=3.14159
7.
```

碁峰 www.gotop.com.tw

```
8.   int main(int argc, char *argv[])
9.   {
10.     float area, circu;                          //宣告浮點數 area,circu
11.     float radius = 5;                           //宣告浮點數 radius=5
12.
13.     area = PI * float(pow(radius, 2));          //area=圓面積 1
14.     circu = 2 * PI * radius;                    //circu=圓周長 1
15.     cout << "圓面積 1 = " << area
16.          << "\t 圓周長 1 = " << circu;
17.     radius = 10;                                //改變 radius=10
18.     area = PI * float(pow(radius, 2));          //area=圓面積 2
19.     circu = 2 * PI * radius;                    //circu=圓周長 2
20.     cout << "\n 圓面積 2 = " << area
21.          << "\t 圓周長 2 = " << circu << endl;
22.     system("PAUSE");
23.     return EXIT_SUCCESS;
24.  }
```

程式輸出

```
圓面積 1 = 78.5397        圓周長 1 = 31.4159
圓面積 2 = 314.159        圓周長 2 = 62.8318
請按任意鍵繼續 . . .
```

2.2.5 C++ 保留字

表 2.5 是 C++ 保留的**關鍵字（keyword）**，包括 C++ 的敘述、參數、函數、或類別等名稱，這些保留字是不能被改變或重新定義的，所以使用者為程式的變數命名時，應避開這些 C++ 的保留字。

表 2.5 C++ 保留字

asm	do	inline	short	typeid
auto	double	int	signed	typename
break	else	long	sizeof	union
bool	enum	mutable	static	unsigned
case	explicit	namespace	struct	using
catch	extern	new	switch	virtual
char	false	operator	template	void
class	float	private	this	volatile
const	for	proteted	throw	while

碁峰 www.gotop.com.tw

continue	friend	public	true	
default	goto	register	try	
delete	if	return	typedef	

2.3 C++ 資料型態

C++ 的**內建資料型態（build-in data type）**包括整數型態、字元型態、浮點數型態、與邏輯型態等。其中整數又分為短整數、整數、與長整數等型態，浮點數又分為單精度、倍精度、與長倍精度等型態。

2.3.1 整數資料 int

整數（integer）以正負號可分為有號數（signed）或無號數（unsigned），以大小可分為短整數（short）、整數（int）與長整數（long），整數變數的宣告型態、儲存空間、與數值範圍如表 2.6 所示。

表 2.6 整數變數的型態、功能、長度、與範圍

宣告型態	宣告功能	儲存空間	數值範圍
short 註 1	短整數	2 bytes	-32,768 至+32,767
unsigned short	無號數短整數	2 bytes	0 至 65,535
signed short	有號數短整數	2 bytes	-32,768 至+32,767
int	整數	4 bytes 註 2	-2,147,483,648 至 +2147483647
unsigned int	無號數整數	4 bytes 註 2	0 至 4,294,967,295
signed int	有號數整數	4 bytes 註 2	-2,147,483,648 至 +2,147,483,647
long 註 1	長整數	4 bytes	-2,147,483,648 至 +2147483647
unsigned long	無號數長整數	4 bytes	0 至 4,294,967,295
Signed long	有號數長整數	4 bytes	-2,147,483,648 至 +2,147,483,647

註 1：short 與 short int 的宣告功能相同，而 long 與 long int 的宣告功能相同。
註 2：16 bits 系統 int 的儲存空間是 2 bytes，32 bits 系統 int 的儲存空間是 4 bytes。

碁峰 www.gotop.com.tw

下面範例是宣告 num0 為整數變數，也就是配置 4 bytes 空間給 num0 變數，因為只宣告 num0 而未指定初值，所以 num0 的初值是未知的，有些系統會自動起始為 0，但有些系統則不會。

```
int num0;       //宣告變數但未給予初值

int num1 = 12345;                                //num1 = 12345
unsigned int num2 = 65432;                       //num2 = 65432
signed int num3 = -12345;                        //num3 = -12345

int short num4 = 12345;                          //num4 = 12345
unsigned int short num5 = 65432;                 //num5 = 65432
signed int short num6 = -12345;                  //num6 = -12345

int long num7 = 1234567890;                      //num7 = 1234567890
unsigned int long num8 = 3210987654;             //num8 = 3210987654
signed int long num9 = -1234567890;              //num9 = -1234567890
```

程式 2-10：宣告整數變數練習

```
1.   //儲存檔名：d:\C++02\C0210.cpp
2.   #include <iostream>
3.   using namespace std;
4.
5.   int main(int argc, char *argv[])
6.   {
7.      signed int num1 = -12345;                //num1 = -12345
8.      unsigned short num2 = 65432;             //num2 = 65432
9.      long num3 = 1234567890;                  //num3 = 1234567890
10.
11.     cout << "有號整數：" << num1 << endl      //輸出字串、數值、跳行
12.          << "無號短整數：" << num2 << endl   //輸出字串、數值、跳行
13.          << "長整數：" << num3 << endl;      //輸出字串、數值、跳行
14.     system("PAUSE");
15.     return EXIT_SUCCESS;
16.  }
```

程式輸出

```
有號整數：-12345
無號短整數：65432
長整數：1234567890
請按任意鍵繼續 . . .
```

碁峰 www.gotop.com.tw

2.3.2 字元資料 char

表 2.7 列出字元與字串變數的宣告型態、宣告功能、儲存空間、與數值範圍。

表 2.7 字元變數的型態、功能、與長度

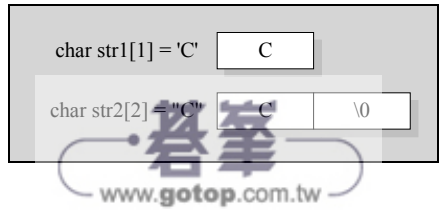
型態	宣告功能	儲存空間	範例
char	宣告字元	1 位元組	char letter = 'C';
char[n]	宣告字串	n 位元組	char str1[3] = {'C', '+', '+'}; char str2[4] = "C++"; char str3[] = "C++ 全方位學習";

註：字元常數使用單引號，字串常數使用雙引號。若省略 n 值，是宣告不定長字串。

char str1[3] = {'C', '+', '+'}; 是配置 3 個字元空間給變數 str1，char str2[4] = "C++"; 是配置 4 個字元空間給變數 str2，char str3[] = "C++ 全方位學習"; 則是根據字串資料的長度配置空間給 str3。第 7 章與第 9 章將會更詳細的介紹陣列與字串的觀念。

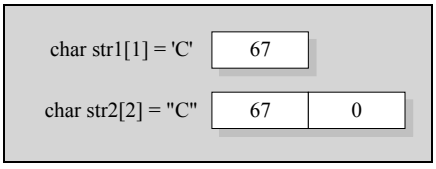
ASCIIZ 字串是以 '\0' 為結尾的字串。當程式使用字串資料時，C++ 會自動加入 '\0' 到**字串**資料的尾端形成 ASCIIZ 字串，但不會加入 '\0' 到**字元**資料的尾端。所以在 C++ 語言中，字元資料（'C'）與字串資料（"C"）是不同的，C++ 將在 "C" 結尾加入 '\0' 如下圖。因此，應配置 3 個字元空間給字元陣列資料 {'C', '+', '+'}，而應配置 4 個字元空間給字串資料 "C++"。

字元資料 'C' 佔用 1 byte 的記憶體空間，而字串資料 "C" 則佔用 2 bytes 的記憶體空間。所以宣告陣列變數 str1 時，只保留 1 byte 的空間，而宣告陣列變數 str2 時，則須保留 2 bytes 的空間如下圖。



www.gotop.com.tw

事實上，C++ 會先將字元資料或字串資料轉成 ASCII 碼再存入記憶體中，所以字元 'C' 會先轉成 A 的 ASCII 碼 67 再存入記憶體中，而 '\0' 會先轉成 ASCII 碼 0 再存入記憶體中，如下圖則是實際存在記憶體中的資料。



下面範例的敘述 char letter0 是宣告字元變數但未指定變數初值，char letter1 = 'C' 則宣告並指定 letter1 的初值為字元 C，char letter2 = 67 的初值 67 等於字元 C 的十進位 ASCII 碼，char letter3 = 0x43 的初值 0x43 等於字元 C 的十六進位 ASCII 碼，char tab = '\t' 是指定 tab 的初值為定位符號，char string[] = "ANSI C++" 的 char string[] 為宣告字元陣列，此字元陣列的初值可以是許多字元或一個字串，請參閱 7.4 節字串陣列說明。

```
char letter0;                          //宣告字元變數但未給予初值
char letter1 = 'C';                    //直接以單引號定義字元
char letter2 = 67;                     //以'C'的ASCII 值定義字元
char letter3 = 0x43;                   //以'C'的16 進位ASCII 值
char tab = '\t';                       //宣告字元並令 tab=定位符號
char string[] = "ANSI C++";            //字串 str="ANSI C++"
```

註：0x43 表示十六進位數，其對應的十進位數為 4*161+3*160=67。

程式 2-11：宣告字元變數練習

```
1.   //儲存檔名：d:\C++02\C0211.cpp
2.   #include <iostream>
3.   using namespace std;
4.
5.   int main(int argc, char *argv[])
6.   {
7.      char str1[] = "第一欄";              //宣告字串 str1="第一欄"
8.      char str2[] = "第二欄";              //宣告字串 str2="第二欄"
9.      char tab = '\t';                     //宣告字元 tab=定位符號
10.     cout << str1 << tab << str2 << endl; //輸出字串、字元、字串
11.     system("PAUSE");
12.     return EXIT_SUCCESS;
13.  }
```

碁峰 www.gotop.com.tw

程式輸出

```
第一欄    第二欄
請按任意鍵繼續 . . .
```

2.3.3 浮點資料 float, double

浮點數可分為單精度、倍精度、與長倍精度浮點數。單精度、倍精度、與長倍精度的差異在數值的有效位數與數值範圍。其中單精度為 7 位有效位數，倍精度與長倍精度為 15 位有效位數。至於浮點變數的宣告型態、儲存空間、與數值範圍如表 2.8 所示。

表 2.8 浮點數變數的型態、功能、長度、與範圍

型態	宣告功能	儲存空間	數值範圍
float	單精度浮點數	4 位元組	$\pm 3.4*10^{-38}$ 至 $\pm 3.4*10^{+38}$
double	倍精度浮點數	8 位元組	$\pm 1.7*10^{-308}$ 至 $\pm 1.7*10^{+308}$
long double	長倍精度浮點數	8 位元組	$\pm 1.7*10^{-4932}$ 至 $\pm 1.7*10^{+4832}$

註：有些編譯器配置 10 個位元組的空間給 long double 型態變數。

下面範例的敘述 float num0 是宣告浮點變數但未指定變數的初值，float pi1 = 3.14159f 是指定 pi1 的初值為 3.14159f，f 表示將常數數值限定為浮點數，若省略 f 則有些編譯器會出現警告訊息。float value2 = 4.5e+16f 指定 value2 等於 $4.5*10^{16}$，double pi3 = 3.141592653 指定 pi3 的初值為 3.141592653，double value4 = 4.5e+101 指定 value4 的初值為 $4.5*10^{101}$。

```
float num0;                          //宣告浮點變數但未給予初值
float pi1 = 3.14159f;                //宣告 pi1 = 3.14159
float value2 = 4.5e+16f;             //宣告 value2 = 4.5*10^16
double pi3 = 3.141592653;            //宣告 pi3 = 3.141592653
double value4 = 4.5e+101;            //宣告 value4 = 4.5*10^101
```

程式 2-12：宣告浮點變數練習

```
1.   //儲存檔名：d:\C++02\C0212.cpp
2.   #include <iostream>
3.   #include <cfloat>
4.   using namespace std;
5.
```

碁峰 www.gotop.com.tw

```
6.   const float PI = 3.14159f;                    //宣告常數 PI = 3.14159
7.
8.   int main(int argc, char *argv[])
9.   {
10.     char str1[] = "單精數 pi = ";              //宣告字串 str1
11.     char str2[] = "倍精數 val = ";             //宣告字串 str2
12.     double val = 4.5e+101;                     //宣告 val = 4.5e+101
13.     cout << str1 << PI << endl                 //輸出字串、單精數、跳行
14.          << str2 << val << endl;               //輸出字串、倍精數
15.     system("PAUSE");
16.     return EXIT_SUCCESS;
17.  }
```

程式輸出

```
單精數 pi = 3.14159
倍精數 val = 4.5e+101
請按任意鍵繼續 . . .
```

2.3.4 邏輯資料 bool

邏輯數值是用來表示運算式的真或假，數值 1 或 true 表示真，數值 0 或 false 表示假。表 2.9 列出邏輯變數的宣告型態、宣告功能與數值範圍。

表 2.9 邏輯變數的型態、功能、長度、與範圍

宣告型態	宣告功能	數值範圍
bool	邏輯變數	true（1）或 false（0）

下面範例的敘述 bool fontBold 是宣告邏輯變數但未指定變數的初值，bool fontitalic = true 則指定 fontItalic 的初值為 true。

```
bool fontBold;                          //宣告邏輯變數但未給予初值
bool fontItalic = true;                 //宣告 fontItalic = 真
```

程式 2-13：宣告邏輯變數練習

```
1.   //儲存檔名：d:\C++02\C0213.cpp
2.   #include <iostream>
3.   using namespace std;
4.
5.   int main(int argc, char *argv[])
6.   {
```

碁峰 www.gotop.com.tw

```
7.      bool bValue = true;                          //宣告並起始邏輯變數值
8.
9.      cout << "邏輯預設值 = " << bValue << endl;   //顯示訊息與邏輯變數值
10.     bValue = false;                              //更改邏輯變數值
11.     cout << "邏輯更改值 = " << bValue << endl;   //顯示訊息與邏輯變數值
12.     system("PAUSE");
13.     return EXIT_SUCCESS;
14.  }
```

程式輸出

```
邏輯預設值 = 1
邏輯更改值 = 0
請按任意鍵繼續 . . .
```

2.3.5 取得型態大小 sizeof

sizeof(資料型態|變數名稱)

- **sizeof** 運算符號（operator）用於取得資料型態如 int、float、bool 的儲存空間或變數的儲存空間。
- **資料型態（data type）**是 C++ 的原始資料型態，也就是 2.3.1 節至 2.3.4 節所介紹的資料型態。也可以是使用者自定的資料型態，本書後面章節將會介紹使用者自定的資料型態。
- **變數名稱（variable）**是宣告過的變數名稱。

程式 2-14：取得資料型態大小練習

```
1.   //儲存檔名：d:\C++02\C0214.cpp
2.   #include <iostream>
3.
4.   using namespace std;
5.
6.   int main(int argc, char *argv[])
7.   {
8.      double dType;
9.
10.     cout << "int 型態的位元組數 = "
11.          << sizeof(int) << "bytes\n";          //取得 int 型態大小
12.     cout << "short 型態的位元組數 = "
```

碁峰 www.gotop.com.tw

```
13.          << sizeof(short) << "bytes\n";    //取得 short 型態大小
14.     cout << "bool 型態的位元組數 = "
15.          << sizeof(bool) << "bytes\n";     //取得 bool 型態大小
16.     cout << "變數 dType 的位元組數 = "
17.          << sizeof(dType) << "bytes\n";    //取得變數 dType 的大小
18.     system("PAUSE");
19.     return EXIT_SUCCESS;
20.  }
```

程式輸出

```
int 型態的位元組數 = 4bytes
short 型態的位元組數 = 2bytes
bool 型態的位元組數 = 1bytes
變數 dType 的位元組數 = 8bytes
請按任意鍵繼續 . . .
```

2.4 習題

選擇題

1. C++ 敘述的結束符號是_____。

   a) 句號（.）　b) 冒號（:）　c) 分號（;）　d)右大括號（ ）

2. C++ 形式註解的起始符號是_____。

   a) \\　b) //　c) *\　d) */

3. 在 cout 敘述輸出，_____是插入新行（newline）的字元。

   a) '\t'　b) '\n'　c) '\f'　d) '\r'

4. 在 cout 敘述輸出，_____是跳至下一定位點（tab）的字元。

   a) '\t'　b) '\n'　c) '\f'　d) '\r'

5. 宣告常數符號時，必須指定_____。

   a)資料型態　b) 常數符號　c) 初值　d) 以上皆對

6. _____是有效的變數名稱。

   a) tri-Area　b) tri Area　c) tri_Area　d) 123Area

碁峰 www.gotop.com.tw

7. _____是有效的短整數（short）資料。

a) 65,536　　b) -145,694　　c) 5,621　　d) +65

8. _____是有效的整數（int）資料。

a) 5.14326　　b) 514326　　c) 5.14326f5　　d) 以上皆對

9. _____不是 1 byte 的字元（char）資料。

a) 'A'　　b) "A"　　c) 65　　d) 0x41

10. 科學記號 4.92143×10^5 的 C++ 形式是_____。

a) 4.92143d5　　b) 4.92143e5　　c) 4.92143f5　　d) 4.92143g5

實作題

11. 寫一程式以 7×6 星號（*）矩陣顯示一個矩形如下。

```
******
******
******
******
******
******
******
```

12. 寫一程式以 7×6 星號（*）矩陣顯示一個菱形如下。

```
*
***
*****
*******
*****
***
*
```

13. 寫一程式以 7×6 星號（*）矩陣顯示一個字母 B 如下。

```
*****
*    *
*    *
*****
*    *
*    *
*****
```

14. 寫一程式以 7×6 星號（*）矩陣顯示自己姓氏的英文縮寫。

15. 寫一程式先定義與起始個人的姓名、電話、與地址等變數，然後將變數內容顯示於視窗中。

碁峰 www.gotop.com.tw

```
你（妳）的姓名
你（妳）的電話
你（妳）的地址
```

16. 寫一程式取得下列變數的大小，並顯示於視窗中。

```
char[] bookName = "C++ 全方位學習";
```

17. 寫一程式顯示庫存管理清單如下。

```
1. 庫存管理
2. 顯示清單
3. 新增項目
4. 更新紀錄
5. 刪除項目
6. 結束
選擇 (1 - 6)：
```

18. 寫一程式顯示庫存清單如下。

```
庫存清單
日期：
編號    名稱    進貨    出貨    庫存
-----------------------------------------------------------------

-----------------------------------------------------------------
```

19. 寫一程式顯示全班成績表的名稱、日期、標題、分格線與總平均如下。

```
全班成績表
日期：
姓名    國文    英文    數學    總分
-----------------------------------------------------------------

-----------------------------------------------------------------
總平均
前 1/4 平均
前 1/2 平均
```

碁峰 www.gotop.com.tw