

C++資料結構與程式設計

C++ 整合開發環境介紹

NTU CSIE 張傑帆

C++的程式架構

C++是由C延伸出來的，因此C++當然包含C語言的所有功能，另外提供完整物件導向程式設計(Object-Oriented Programming：OOP)功能。

C++可說是C語言的加強版。

C++的程式架構和C語言很類似，而且程式的進入點都是由main()函式開始。

新式標頭

C程式中若有使用到`printf()`和`scanf()`輸出入函式時，由於這些函式的宣告都在`stdio.h`標頭檔內，因此必須先將此標頭檔含入到程式的最前面，寫法如下：

```
#include<stdio.h>
```


新式標頭

C++逐漸取代傳統的C語言，為了能向下與C語言相容，標準的ANSI/ISO C++(簡稱標準C++)仍支援傳統C語言的標頭檔。

只不過在C++程式中，若有使用到這些C語言所提供的標頭檔時，我們建議使用新式標頭寫法。

其做法就是在標頭檔名稱最前面加上小寫的c和省略副檔名*.h即可。例如：

```
#include<cstdio>
```

標準C++為了提升功能，引進新的C++標準程式庫函式，當然也使用新式標頭檔寫法，以和ANSI C有所區別。

由於新式標頭不是檔名而是一個標頭名稱，是由識別字組成，因此在含入新式標頭名稱時不要加上.h副檔名。

下列即是標準C++ 所提供一些新式標頭名稱寫法：

```
#include<iostream>
#include<string>
#include<fstream>
#include<list>
#include<vector>
#include<map>
```

輸入/出資料

標準的C++程式，則是使用`cout`和`cin`物件來輸出入資料。

由於這兩個物件的宣告都定義在C++標準程式庫中的`iostream`。

在C++ 的程式中，若有使用到`cin`或`cout`物件，都必須在程式最前面先將`iostream`含入進來。寫法如下：

```
#include<iostream>
```


C++寫一個簡單cout程式，將 “Hello,” 和
“這是第一個C++程式” 兩個字串分別用cout逐行輸出：

Hello,
這是第一個C++程式

// Program : greeting.cpp ← 註解開頭

include <iostream> ← 程式中使用到 cout 必須含入此標題名

int main(int argc, char *argv[]) ← 程式由此開始執行，int 表示傳回值的資料型別為整數

{

std::cout << "Hello,\n "; ← 輸出雙引號括住的字串常數，並將游標移到下一行的開頭

std::cout << "這是第一個 C++ 程式." << std::endl;

system("PAUSE");

return 0; ← 將 0 傳回給系統表示系統正常結束
若傳回非零值代表失敗

}

命名空間

已標準化後的C++，標頭檔的所有內容放在std命名空間中，因此程式中使用到cout或cin時，就必須如上面敘述在前面加上 std::，此種寫法造成撰寫上的不方便。

為解決此問題C++ 提供using directive方式，即在程式之前加上 `using namespace std;` 敘述把std完全打開，讓std命名空間變得可見，如此上面範例在程式中便可將 std:: 拿掉。

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    cout << "Hello,\n ";
```

```
    cout << "這是第一個 C++程式." <<endl;
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

命名空間最主要的功能

可以避免不同區域的識別字彼此互相衝突

例如：C語言所提供程式庫內的函式名稱我們是無法再重新定義，也就是說我們無法使用C語言再重新定義一個printf()函式

但是在C++ 的程式中允許使用namespace自定的mynamespace命名空間來重新定義一個printf()函式，而若要呼叫此函式可以寫為 mynamespace::printf();



按一下「注
以建立、標
案」。

```
01 #include <cstdlib>
02 #include <iostream>

03 using namespace std;      // 引用 std 命名空間
04
05 namespace mynamespace // 宣告 mynamespace 為自定命名空間
06 {
07     void printf()          // 在 mynamespace 命名空間定義 printf()函式
08     {
09         cout << "C++的 printf()函式\n";
10     }
11 };

12 int main(int argc, char *argv[])
13 {
14     printf("C 語言的 printf()函式\n");// 呼叫 C 語言標準程式庫中 printf()函式
15     mynamespace::printf();           // 呼叫 mynamespace 下的 printf()函式
16     system("PAUSE");
17     return 0;
18 }
```

上述程式的執行結果會印出：

C語言的printf()函式

C++的printf()函式

單行註解與多行註解

[例] 單行註解寫法

① `// 這是註解`

② `int i,k; // 這是註解`

[例] 多行註解寫法

`/*`

第 1 行註解

第 2 行註解

⋮

第 n 行註解

`*/`