

程式基本架構 與輸出

林靖紳

Contents

- 1 What's Programming
- 2 C++ 程式架構
- 3 Output
- 4 程式編譯、執行錯誤分類

What's Programming

01

What's Programming



What's Programming

範例問題

王老師在學期末打算核對學生的期末總成績，其計算方式為平時測驗 20 %、上課表現 20 %、期中考 30%、期末考 30%。但王老師有很多的學生，不太可能一一進行手動計算，因此請你寫出一個程式，幫助王老師完成這個任務。

輸入說明

輸入資料一開始為一個正整數 n ，代表這個班上有 n 位學生。

接下來有 n 行，每一行有 4 個正整數，分別代表每一位學生的平時測驗、上課表現、期中考及期末考分數。

$$2^{31} - 1 \geq n \geq 0$$

輸出說明

請依序輸出每一位學生的期末總成績

C++ 程式架構

02

C++ 程式架構

最基本的架構

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      // 主要程式碼，像是：
8
9      cout<<"Hello world"<<endl;
10     return 0;
11 }
12
```

C++ 基本概念：

- 程式碼是由上而下執行的
- 在 main 裡面直到遇到 return 0；程式會立刻結束
- 程式中用到任何物件都需要進行宣告 (下一章)

C++ 程式架構

最基本的架構

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      // 主要程式碼，像是：
8
9      cout<<"Hello world"<<endl;
10     return 0;
11 }
12
```

- #include 標頭檔 (header)
- 標頭檔 = (函式)s
 - 是一個存放 [已經定義好的函式] 的文件
- 公式 $\left\{ \begin{array}{l} \text{\color{red}\code{\#include <標頭檔名稱>}} \text{ 適用於官方定義好的函數庫 (標準函數庫)} \\ \text{\code{\#include "標頭檔名稱"}} \text{ 適用於自定義的函數庫} \end{array} \right.$
- <iostream> 代表資料串流輸入與輸出 (Input/Output Stream) 的標頭檔，像是
 - cin, cout

C++ 程式架構

最基本的架構

```
1
2
3  using namespace std;
4
5  int main(){
6
7      // 主要程式碼，像是：
8
9      cout<<"Hello World"<<endl;
10     return 0;
11 }
```

- 如果將 `#include` 拿掉之後編譯會如何呢？

C++ 程式架構

最基本的架構

- 會得到 **Compile Error** (參見 Part 04)

```
PS C:\Users\tiffany\Desktop> g++ week1_test.cpp -g -o week1_test.exe
week1_test.cpp: In function 'int main()':
week1_test.cpp:9:5: error: 'cout' was not declared in this scope
   9 |     cout<<"Hello World"<<endl;
     |     ^~~~~
week1_test.cpp:1:1: note: 'std::cout' is defined in header '<iostream>'; did you forget to '#include <iostream>'?
+++ |+#include <iostream>
   1 |
week1_test.cpp:9:26: error: 'endl' was not declared in this scope
   9 |     cout<<"Hello World"<<endl;
     |                        ^~~~~
week1_test.cpp:1:1: note: 'std::endl' is defined in header '<ostream>'; did you forget to '#include <ostream>'?
+++ |+#include <ostream>
   1 |
```

C++ 程式架構

最基本的架構

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      // 主要程式碼，像是：
8
9      cout<<"Hello world"<<endl;
10     return 0;
11 }
12
```

-
- C++ 標準函式庫被定義於一個名為 `std` 的 namespace 中
 - 包含 `<iostream>`
 - 像是 `std::cout<<" Hello World" <<std::endl;`
 - 使用 `using namespace std` 可以省去接下來使用這些識別符號時前面都要加上 `std::`

C++ 程式架構

最基本的架構

```
1  #include <iostream>
2
3
4
5  int main(){
6
7      // 主要程式碼，像是：
8
9      cout<<"Hello World"<<endl;
10     return 0;
11 }
```

- 如果我們把 `using namespace std` 拿掉後編譯會如何呢？

C++ 程式架構

最基本的架構

- 會得到另一個 **Compile Error** !

```
PS C:\Users\tiffany\Desktop> g++ week1_test.cpp -g -o week1_test.exe
week1_test.cpp: In function 'int main()':
week1_test.cpp:9:5: error: 'cout' was not declared in this scope; did you mean 'std::cout'?
    9 |     cout<<"Hello World"<<endl;
      |     ^~~~~
      |     std::cout
In file included from week1_test.cpp:1:
c:\mingw\lib\gcc\mingw32\9.2.0\include\c++\iostream:61:18: note: 'std::cout' declared here
   61 |     extern ostream cout;   /// Linked to standard output
      |                  ^~~~~
week1_test.cpp:9:26: error: 'endl' was not declared in this scope; did you mean 'std::endl'?
    9 |     cout<<"Hello World"<<endl;
      |                          ^~~~~
      |                          std::endl
In file included from c:\mingw\lib\gcc\mingw32\9.2.0\include\c++\iostream:39,
      from week1_test.cpp:1:
c:\mingw\lib\gcc\mingw32\9.2.0\include\c++\ostream:599:5: note: 'std::endl' declared here
  599 |     endl(basic_ostream<_CharT, _Traits>& __os)
      |     ^~~~~
```

C++ 程式架構

最基本的架構

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6
7      // 主要程式碼，像是：
8
9      cout << "Hello world" << endl;
10     return 0;
11 }
12
```

- `int main()` 宣告主函式
- 在每一支程式中，**僅能宣告 1 個 `main` 函式**
- 其中 `{ }` 裡面會放主要的程式碼
- 也可能會看到 `int main (int argc, char *argv[])`
- 而在 **C 語言** 中可能也會看到 `void main(){ }`
 - 但是在 C++ 中此種寫法不太相容

C++ 程式架構

最基本的架構

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      //主要程式碼，像是：
8      //我是一行註解
9      /*
10         我
11         是
12         一
13         排
14         註
15         解
16     */
17     cout<<"Hello world"<<endl;
18     return 0;
19 }
```

- 「//」開頭表示本行為註解
- 註解可以增加程式的可閱讀性，讓別人了解你的程式的功能
- /* */ 表示「/*」到「*/」區間內的所有文字都是註解
- 盡量還是使用 英文 作註解，中文很常會因為環境問題造成編譯問題

C++ 程式架構

最基本的架構

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      // 主要程式碼，像是：
8
9      cout<<"Hello world"<<endl;
10     return 0;
11 }
12
```

- ；為每一句程式碼的結尾，可以想像成每一句話結束時的。
- 在描述完每一行程式碼時，我們都需要加上；

C++ 程式架構

最基本的架構

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6
7      // 主要程式碼，像是：
8
9      cout<<"Hello world"<<endl;
10     return 0;
11 }
12
```

- 在 main 裡面放 return 0; 表示此程式碼結束
- return 本身的作用是中止函數的運行，後面可以加上函式的返回值
- 0 這個返回值對於 main 函式來說，是告訴系統程式執行完了，並且沒有問題，正常退出此程式

Output 03

Output

輸出文字

- 格式: `cout << 變數或是字串1<<變數或是字串2<<變數或是字串3<<....<<變數或是字串n` ;
- 如果要輸出字串(一行文字), 需要利用 “ ” 來框起來, 像是:
 - `cout << “Hello !” << “Hi” ;`

```
PS C:\Users\tiffany\Desktop> ./week1_test.exe
Hello World!Hi
```

- 而 `endl` 表示 end of line, 結束一行, 輸出後換行, 像是:
 - `cout << “Hello!” <<endl<< “Hi” <<endl;`

```
PS C:\Users\tiffany\Desktop> ./week1_test.exe
Hello World!
Hi
```

小提醒:

- 輸出文字盡量還是使用 英文, 中文很常因為環境問題造成編譯問題

[練習一]

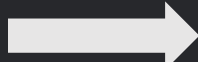
Output

輸出文字

- 這時候，內心應該要冒出一個問題
 - 那如果我想輸出 “ 怎麼辦呢？ ”
 - 試試 `cout << “ ” ;`

Output

輸出文字

- 這時候，內心應該要冒出一個問題
 - 那如果我想輸出 “ 怎麼辦呢？
 - 試試 `cout << “ ” ;`  會發現得到 Compile error

```
PS C:\Users\tiffany\Desktop> g++ week1_test.cpp -g -o week1_test.exe
week1_test.cpp:6:13: warning: missing terminating " character
   6 |     cout<<""<<endl;
      |               ^
week1_test.cpp:6:13: error: missing terminating " character
   6 |     cout<<""<<endl;
      |               ^~~~~~
week1_test.cpp: In function 'int main()':
week1_test.cpp:6:13: error: expected ';' before 'return'
   6 |     cout<<""<<endl;
      |               ^
      |               ;
   7 |
   8 |     return 0;
      |     ~~~~~
```

Output

輸出文字

- 跳脫字元
 - 指的是脫離原本字元的意思
 - 加上 \ 用以跳脫，有的字元本身在程式碼中有特殊的用途，需要用 \ 跳脫；
 - 有些則是加上 \ 之後會有脫離原本字元意思的用途
 - 例如前頁的例子中，「”」用來當成 C++ 字串的開頭或是結尾
 - 因此我們需要加上「\」變成「\"」就能在字串中當作純符號來看待
- 試試看 `cout << “\” << endl;`

Output

輸出文字

- 跳脫字元

• 試試看 `cout << "\\\" <<endl;`  成功了！

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      cout<<"\\"<<endl;
7      return 0;
8  }
9
```

```
PS C:\Users\tiffany\Desktop> g++ week1_test.cpp -g -o week1_test.exe
PS C:\Users\tiffany\Desktop> ./week1_test.exe
"
```

Output

輸出文字

- 跳脫字元
- 其他常用的跳脫字元

字元	意義
\'	單引號
\"	雙引號
\\	反斜線
\0	空字元(NULL)
\t	定位字元(TAB)
\n	換行字元(ENTER)

→ 輸出單引號

→ 輸出雙引號

→ 輸出反斜線

→ 表示字串的結尾 ——> 他很特別，之後會來介紹

→ 輸出 tab 的空格 (4格或8格)

→ 輸出換行 (類似 endl)

Output

輸出文字

- 跳脫字元 – 舉例 `\n`

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      cout<<"Hello World\nHi"<<endl;
7
8      return 0;
9  }
```

```
PS C:\Users\tiffany\Desktop> g++ week1_test.cpp
PS C:\Users\tiffany\Desktop> ./week1_test.exe
Hello World
Hi
```

Output

輸出文字

- 跳脫字元 – 舉例 \t

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      cout<<"Hello World\tHi"<<endl;
7
8      return 0;
9  }
```

```
PS C:\Users\tiffany\Desktop> g++ week1_test.cpp
PS C:\Users\tiffany\Desktop> ./week1_test.exe
Hello World      Hi
```

[練習二]

程式編譯、執行錯誤分類

04

分類

- AC
 - Accept, 表示完全答對! 從編譯到輸出結果和測試都是對的
- WA
 - Wrong Answer, 表示編譯成功, 但是測試有出現錯誤
- CE
 - Compile Error, 表示編譯不成功
- TLE
 - Time Limit Exceeded, 表示編譯成功但是程式執行花費太多時間
- RE
 - Runtime Error, 會出現這種狀況有很多種原因, 最常見的情況可能是因為溢位(overflow)
 - 這是一個很嚴重的 Error, 需要從頭開始好好審查自己的 code

Thank You