

資訊奧林匹亞講義(1)

C++內建

基本規則：

- (1) 基本上在每一行的最後都要加分號，如果你多加或少加了在編譯的時候程式會告訴你，但有時候錯誤會顯示在下一行就是了。
- (2) 兩條斜線 (//) 後面的東西是註解，程式不會執行註解的東西，註解看似沒有任何用處，但他對提醒寫程式的人很有幫助。如果要註解的東西不只一行的話要使用 /* */ 框住想要註解的東西。

範例：

```
// 單行註解
/*
    多行註解
    多行註解
*/
```

- (3) 大部分的時候都可以隨意的換行或空格，不會影響程式的執行，但適當的換行及空格有助於增加你的程式的“可讀性”。

變數：

- (1) int：int 指的是整數，具體來說它可以用來儲存 -2147483648~2147483647 之間的數字。儲存的範圍通常是足夠正常使用的，但是如過超出這個範圍程式就會產生一些不可預期的錯誤，當找不到bug在那裡的時候可以看看是不是超出範圍了。

範例：

```
int n; // 向程式宣告一個整數 n。
int n,m; // 向程式宣告兩個整數，一個叫做 n，另一個叫做 m。
int n=3; // 向程式宣告一個整數 n，並設定 n 為 3。
```

- (2) double：double 的使用方式和 int 一模一樣，只不過他所宣告的是一個可以有小數點的數字，我們稱呼這個可以有小數點的數字為“浮點數”。

範例：

```
double n; // 向程式宣告一個浮點數 n。
double n,m; // 向程式宣告兩個浮點數，一個叫做 n，另一個叫做 m。
double n=3; // 向程式宣告一個浮點數 n，並設定 n 為 3。
```

- (3) char：char 和前面都一樣，而宣告的東西是一個“字元”，可以是一個英文字母、一個數字 (0~9)，又或是特殊符號(!@#\$\$%^&* 之類的)，其中比較特殊的是 \n 是換行，\0 是沒有東西。

範例：

```
char c; // 向程式宣告一個字元 c。
char c='a'; // 向程式宣告一個字元 c，並設其為 a。
char c[50]; // 向程式宣告50個字元，他們是 c[0] 到 c[49]，總稱為 c。
char c[50]="sam"; // 向程式宣告50個字元，並在裡面存入“sam”。
// 宣告一堆字元的部分比較複雜，之後會做詳細的解釋，這邊會用就好
// “字元”指的是一個字，像‘a’，‘1’等等的都叫字元
```

// “字串”指的是一串字，像 “anna”，“13459” 都是字串
// 這邊要注意的是 ‘ ’ 裡面包的是一個 “字元”，而 “ ” 裡面包的是一個 “字串”

運算：

- (1) + - * / ：和數學的 + - * / 一樣，這邊要注意如果是 int(整數) 在做除法的時候會無條件捨去，如果想要小數點的話要先轉成double。

範例：

```
double a ;  
int b=5,c=3 ;  
a = b / c ; // 這邊 a 會是 1 。  
a = (double) b / (double) c ; // 這邊 a 會是 1.666 。  
// 其實好像 a = (double) b / c 就可以了，不過保險起見兩個都轉換型態也不會怎麼樣。
```

- (2) =：這邊就和數學的 = 不一樣了，他的意思是“賦值”，比如說 a = a + b，是把 a+b 這個值存給 a。

範例：

```
a = b + c ; // 把 b + c 的值存到 a
```

- (3) %：除法中的餘數，比方說 7%5 = 2。

- (4) +=, -=, *=, /=, %=：a += b 跟 a = a + b 是一樣的意思，只是前者寫起來比較快。

- (5) ++, --：a++ 跟 a += 1 是一樣的意思，只是寫起來比較快。

- (6) ==, &&, ||：因為 = 是賦值的意思，所以如果要比較是否相等的話要用 ==。同樣的，因為 & 有別的意思，所以要表示 “而且” 的意思的時候要用 &&。|| 也是同理，這邊是 “或著” 的意思。

- (7) >, <, >=, <=：跟數學學的東西一樣。

判斷、迴圈：

- (1) if(){ }： “if” 在中文的意思就是 “如果”。在小括號裡面會放入要判斷的條件，大括號裡面放入要做的事。

範例：

```
if( 超過十點了 ){ 睡覺; } // 如果超過十點了就去睡覺，這邊要注意寫中文只是方便理解而已，照抄的話會無法執行。  
if( bmi>=24 )printf(“you are fat!!\n”);  
if( bmi<=16 )printf(“you are thin!!\n”);  
// 如果大括號裡面的東西只有一個的話可以省略大括號。
```

- (2) else{ }：需要接在 if(){ } 的後面，中文的意思是 “否則”。

範例：

```
if( score>=60 )printf(“passed\n”);  
else printf(“failed\n”);  
// 一樣，如果大括號裡面如果只有一個東西的話可以省略大括號。
```

- (3) else if(){ }：可以理解成 else{ if(){ } }。

範例：

```
if( number>0 )printf(“positive\n”);  
else if( number<0 )printf(“negative\n”);  
else printf(“zero\n”);  
//跟下面的意思一模一樣
```

```

if( number>0 )printf("positive\n");
else{
    if( number<0 )printf("negative\n");
    else printf("zero\n");
}

```

- (4) while(){ }：中文的意思是“當...的時候，做...”。跟 if 不一樣的地方是當條件符合的時候他會一直重複。

範例：

```

int number = 5 ;
while( number>0 ){
    number--;
    printf("hello\n");
}

```

// 這樣子程式就會印出hello 5次

// 同理如果大括號裡面只有一個東西的話可以省略

- (5) for(; ;){ }：這是回圈裡面最複雜的一個。他一共要填入4個東西，變成 for(A; B; C){ D } 程式在執行的時候會照這個順序執行：ABDCBDCBDC....C

範例：

```

for(int i = 0; i<5; i++)printf("hello\n"); // 這段程式會印出5次hello

```

#include <stdio.h>

- (1) printf(".....")：printf 是 print + f(function 函式)，顧名思義，他是一個可以幫助你讓程式印出東西的函式。

範例：

```

printf("abcd1234"); // 程式會印出 abcd1234
printf("Hello World!\n"); // 程式會印出 Hello World!
/*

```

這邊 \n 是換行的意思，那如果你想印出 \n 的時候，要在前面加一個 \ 變成 printf("\\n");

同理，當你有像要印出的東西但是他已經有別的意思的時候在前面加一個 \ 就可。

```

*/
int n=5;
char c[50]="hello";
double a=3.3;
printf("n = %d\n",n); // 程式會印出 n = 5
printf("c = \"%s\"\n"); // 程式會印出 c = "hello"
printf("a = %lf\n"); // 程式會印出 a = 3.3

```

// 有時候你要印出的東西不是固定的，可能跟程式執行中的變數有關

- (2) scanf(".....")：跟printf一樣，scan + f(function 函式)，可以讓程式讀入所輸入的資料。

範例：

```

scanf("%d",&n); // 讀入一個整數，並存到 n，要注意如果沒有先宣告 n 程式會錯誤。
scanf("%s",c); //讀入一個字串並存到c，讀字串的時候比較特別，不需要&。

```