

IM1003: Programming Design, Spring 2017

Lab 02

Jhih-Bang Hsieh
bigelephant29

National Taiwan University

Outline

- if-else
- Logical Operators
- switch-case
- while, do-while
- for
- Nested Structure
- Practice

if-else

- 三個組成元件：if、else if、else
- if 為一切的開頭
- else if 只能接在 if 或 else if 後面
- else 只能接在 if 或 else if 後面
- 一旦寫了 else，後面就不能再接任何條件

- 如果今天是晴天，帥哥哥會想出去運動。(if)
- 否則如果今天是陰天，帥哥哥會想出去逛街。(else if)
- 否則的話，帥哥哥會想宅在家裡打 LOL。(else)

if-else

- 你不會劈頭就說「否則」，同理 else if、else 不會當作條件式的開頭。
- 你可以一直有各種特例，所以有很多「否則如果（else if）」是很合理的。
- 最後一個否則，如果沒有帶有條件，則囊括所有其他例外。
- 囊括所有例外的否則（else）最多只有一個。

Logical Operators

- Logical and: `&&`
- Logical or: `||`
- Logical not: `!`
- Unary Operator (單元運算子): `!`
- Binary Operator (二元運算子): `||`、`&&`
- 不確定運算優先順序的話，記得加個括號！

switch-case

```
1  switch(operand) {  
2      case value_1:  
3      case value_2:  
4          // do something  
5      break;  
6      case value_3:  
7          // do something  
8      break;  
9      default:  
10         // otherwise, do something  
11 }
```

- case 跟 case 之間記得要 break。
- 多條件可以省略 break，如範例的 value_1、value_2。

switch-case

```
1  switch(n) {  
2      case 1:  
3      case 2:  
4          int tmp;  
5          break;  
6      case 3:  
7          int tmp;  
8          break;  
9      default:  
10         int tmp;  
11 }
```

- error: redefinition of 'tmp'
- 因為所有 case 目前屬於同一個 block，所以這樣寫是不好的。
- 盡量避免在 switch-case 裡面宣告變數。
- 不得已的時候，請加上大括號！

switch-case

```
1  switch(n) {  
2      case 1:  
3      case 2:  
4      {  
5          int tmp;  
6      }  
7          break;  
8      case 3:  
9      {  
10         int tmp;  
11     }  
12         break;  
13     default:  
14     {  
15         int tmp;  
16     }  
17 }
```


while, do-while

- do-while 非常少用到，但是有些時候它非常好用。
- 無論任何條件下，至少必須執行一次的 while 迴圈。
- 大家可以回想一下 Lab01 Practice B !

for

- 帶有計數器（counter）的迴圈。
- 使用時必須清楚了解 for 的執行順序！

for

```
1  for ( (1) ; (2) ; (3) )  
2  {  
3      (4)  
4  }
```

- ❶ 執行 (1) 進行初始化，在這裡可以宣告暫時變數。
- ❷ 判斷 (2) 條件是否成立，成立則繼續，否則離開。
- ❸ 執行 (4)。
- ❹ 執行 (3)。
- ❺ 回到第 2 點。

for

```
1  for(int i = 0; i < n; i++) {  
2      cout << i << endl;  
3  }
```

```
1  int i;  
2  for(i = 0; i < n; i++) {  
3      cout << i << endl;  
4  }
```

Nested Structure

- 剛剛介紹的所有結構，都可以寫成巢狀結構。(好潮ㄟ)
- 巢狀結構賦予程式碼更多彈性，可以組成更強、更簡潔的邏輯架構。

```
1  for(int i = 0; i < n; i++) {  
2      for(int j = 0; j < m; j++) {  
3  
4      }  
5  }  
6  
7  if(n < 100) {  
8      if(n < 10) {  
9  
10     } else {  
11  
12     }  
13 }
```

Practice

請以本週所學，實作上週的 Practice A、B、C、D。

Practice E

請用巢狀結構輸出九九乘法表，沒有格式限制。