



03 流程控制 (II)

Fundamental Computer Programming- C++ Lab(I)

元智大學 | C++ 程式設計實習

張維元

課程投影片：[請從元智個人 Portal 下載](#)

C++ 程式設計實習

- 01 程式設計與開發環境
- 02 變數型態與運算
- 03 流程控制
- 04 陣列與向量
- 05 函數與遞迴函數
- 06 字串與指標
- 07 循序檔案和隨機檔案

2015



維元

(@v123582)

Web Development

Data Science

#遠端 #斜槓 #教學
#資料科學 #網站開發

擅長網站開發與資料科學的雙棲工程師，熟悉的語言是 Python 跟 JavaScript。同時也是 資料科學家的工作日常 粉專及 資料科學家的 12 堂心法課 發起人，擁有多次國內大型技術會議講者經驗，持續在不同的平台發表對 #資料科學、#網頁開發 或 #軟體職涯 相關的分享。

- 元智大學 C++/CPE 程式設計課程 講師
- ALPHACamp 全端 Web 開發 / Leetcode Camp 課程講師
- CUPOY Python 網路爬蟲實戰研習馬拉松 發起人
- 中華電信學院 資料驅動系列課程 講師
- 工研院 Python AI 人工智慧資料分析師養成班 講師
- 華岡興業基金會 AI/Big Data 技能養成班系列課程 講師

site: v123582.tw / line: weiwei63

mail: weiyuan@saturn.yzu.edu.tw

2015



維元

(@v123582)

Web Development

Data Science

#遠端 #斜槓 #教學
#資料科學 #網站開發

擅長網站開發與資料科學的雙棲工程師，熟悉的語言是 Python 跟 JavaScript。同時也是 資料科學家的工作日常 粉專及 資料科學家的 12 堂心法課 發起人，擁有多次國內大型技術會議講者經驗，持續在不同的平台發表對 #資料科學、#網頁開發 或 #軟體職涯 相關的分享。

- 2018 總統盃黑客松 冠軍隊伍
- 2017 資料科學愛好者年會(DSCONF) 講師
- 2017 行動科技年會(MOPCON) 講師
- 2016 微軟 Imagine Cup 台灣區冠軍

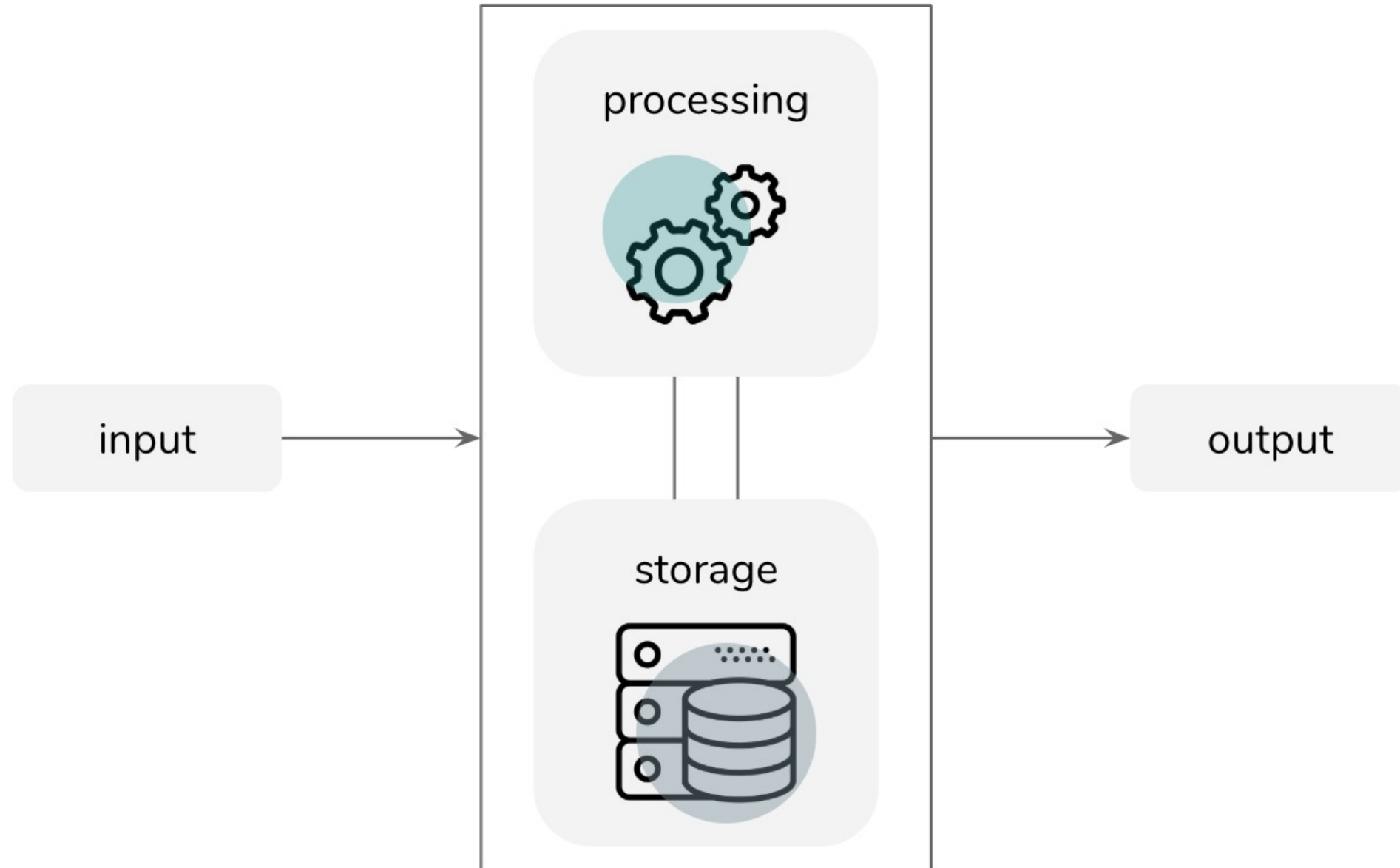
site: v123582.tw / line: weiwei63
mail: weiyuan@saturn.yzu.edu.tw

什麼是程式？

程式語言是用來命令電腦執行各種作業的工具，是人與電腦溝通的橋樑。當電腦藉由輸入設備把程式讀入後，會儲存在主記憶體內，然後指令會依序被控制單元提取並解碼或翻譯成電腦可以執行的信號，並把信號送到各個裝置上，以執行指令所指派的動作。也就是說，人類與電腦溝通的語言稱為程式語言。

程式 = 利用一系列的指令告訴電腦如何執行工作





作業 #08

■#練習：請撰寫一個程式計算 $1! + 2! + 3! + \cdots + n! = ?$

■Requirements：

1. 輸入一個正整數代表 n
2. 請輸出 $1! + 2! + 3! + \cdots + n!$ 的結果

■Sample Input：4

■Sample Output：33

■Hint & Bonus：只有使用一個迴圈

利用迴圈計算總和和階乘

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     int s = 0;
8
9     for (int i = 1; i <= n; i++){
10         s += i;
11     }
12
13     cout << s << endl;
14     return 0;
15 }
16
17
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     int f = 1;
8
9     for (int i = 1; i <= n; i++){
10         f *= i;
11     }
12
13     cout << f << endl;
14     return 0;
15 }
16
17
```


先將題目簡化成 $1 + 2 + \cdots + n$

```
1
2
3 #include <iostream>
4 using namespace std;
5
6 int main(){
7     int n, s = 0, f = 1;
8     cout << "請輸入一個數字 : ";
9     cin >> n;
10    for (int i = 1; i <= n; i++) {
11        s += i;
12    }
13
14    cout << "1! + 2! + ... + n! = " << s;
15    return 0;
16 }
17
18
```

將每一回合的 i 轉化成 f

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int n, s = 0, f = 1;
6     cout << "請輸入一個數字 : ";
7     cin >> n;
8     for (int i = 1; i <= n; i++) {
9         f = 1;
10        for (int j = 1; j <= i; j++) {
11            f *= j;
12        }
13        s += f;
14    }
15
16    cout << "1! + 2! + ... + n! = " << s;
17    return 0;
18 }
```

$n = 4$

$i = 1, f = 1$

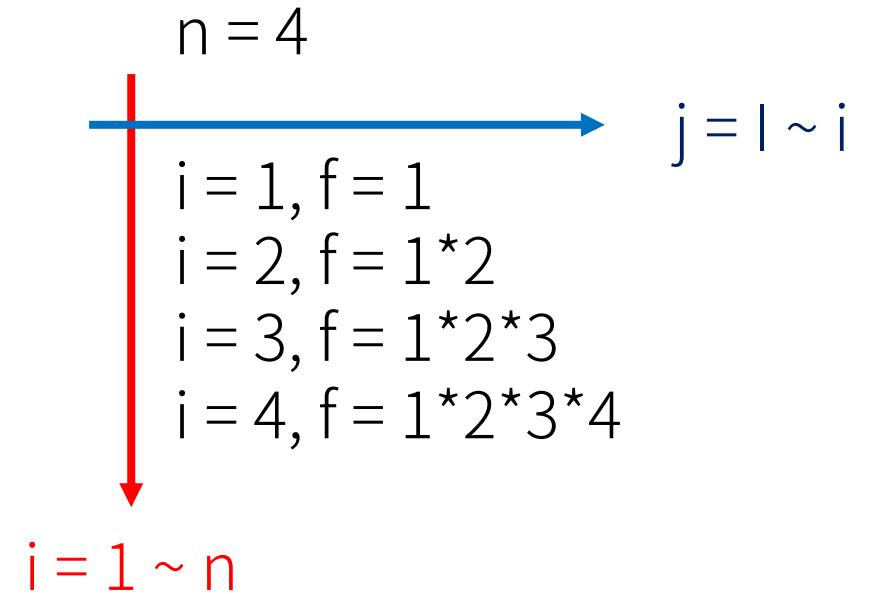
$i = 2, f = 1 * 2$

$i = 3, f = 1 * 2 * 3$

$i = 4, f = 1 * 2 * 3 * 4$

兩層迴圈計算階層總和

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int n, s = 0, f = 1;
6     cout << "請輸入一個數字：";
7     cin >> n;
8     for (int i = 1; i <= n; i++){
9         f = 1;
10        for (int j = 1; j <= i; j++){
11            f *= j;
12        }
13        s += f;
14    }
15
16    cout << "1! + 2! + ... + n! = " << s;
17    return 0;
18 }
```



兩層迴圈計算階層總和

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int n, s = 0, f = 1;
6     cout << "請輸入一個數字：";
7     cin >> n;
8     for (int i = 1; i <= n; i++){
9         f = 1;
10        for (int j = 1; j <= i; j++){
11            f *= j;
12        }
13        s += f;
14    }
15
16    cout << "1! + 2! + ... + n! = " << s;
17    return 0;
18 }
```

$n = 4$

→ $j = 1 \sim i$

$i = 1, f = 1$
 $i = 2, f = 1 * 2$
 $i = 3, f = 1 * 2 * 3$
 $i = 4, f = 1 * 2 * 3 * 4$

每一回合 $1 \times 2 \times \cdots (i-1)$ 可以累計

```
1
2 #include <iostream>
3 using namespace std;
4
5 int main(){
6     int n, s = 0, f = 1;
7     cout << "請輸入一個數字 : ";
8     cin >> n;
9     for (int i = 1; i <= n; i++) {
10         f *= i;
11         s += f;
12     }
13
14     cout << "1! + 2! + ... + n! = " << s;
15     return 0;
16 }
17
18
```

$n = 4$

$i = 1, f = 1$

$i = 2, f = 1! \times 2$

$i = 3, f = 2! \times 3$

$i = 4, f = 3! \times 4$

作業 #09

■#練習：請根據輸入的數值看印出對應高度的菱形圖形：

■Requirements：

1. 輸入一個正整數高度
2. 請輸出符合條件的圖形

■Sample Input：7

■Sample Output：參考右圖

■Bonus：迴圈數量小於等於三個

■Hint：利用 `abs()` 取絕對值

■Note：僅限 10/20、11/03 上課繳交

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

利用迴圈印出圖形

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++ ){
8         for (int j = 1; j <= i; j++ ){
9             cout << "*";
10        }
11        cout << endl;
12    }
13
14    return 0;
15 }
```

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

利用迴圈印出圖形

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++){
8         for (int j = 1; j <= i; j++){
9             cout << "*";
10        }
11        cout << endl;
12    }
13
14    return 0;
15 }
```

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

高度有五行

利用迴圈印出圖形

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++ ){
8         for (int j = 1; j <= i; j++ ){
9             cout << "*";
10        }
11        cout << endl;
12    }
13
14    return 0;
15 }
```

*

→ 第 1 行有 1 個星星

**

→ 第 2 行有 2 個星星

→ 第 3 行有 3 個星星

→ 第 4 行有 4 個星星

→ 第 5 行有 5 個星星

利用迴圈印出圖形

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++){
8         for (int j = 1; j <= n-i; j++){
9             cout << " ";
10        }
11        for (int j = 1; j <= i*2-1; j++){
12            cout << "*";
13        }
14        cout << endl;
15    }
16
17    return 0;
18 }
```

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

利用迴圈印出圖形

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++){
8         for (int j = 1; j <= n-i; j++){
9             cout << " ";
10        }
11        for (int j = 1; j <= i*2-1; j++){
12            cout << "*";
13        }
14        cout << endl;
15    }
16
17    return 0;
18 }
```

```
      *
```

```
    ***
```

```
  *****
```

```
*****
```

```
*****
```

利用迴圈印出圖形

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++){
8         for (int j = 1; j <= n-i; j++){
9             cout << " ";
10        }
11        for (int j = 1; j <= i*2-1; j++){
12            cout << "*";
13        }
14        cout << endl;
15    }
16
17    return 0;
18 }
```

```
      *
```

```
  ***
```

```
*****
```

```
*****
```

```
*****
```

高度有五行

利用迴圈印出圖形

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++){
8         for (int j = 1; j <= n-i; j++){
9             cout << " ";
10        }
11        for (int j = 1; j <= i*2-1; j++){
12            cout << "*";
13        }
14        cout << endl;
15    }
16
17    return 0;
18 }
```

→ 4 個空白 1 個星星

→ 3 個空白 3 個星星






→ 2 個空白 5 個星星

→ 1 個空白 7 個星星

→ 0 個空白 9 個星星

利用迴圈印出圖形






```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++){
8         for (int j = 1; j <= n-i; j++){
9             cout << " ";
10        }
11        for (int j = 1; j <= i*2-1; j++){
12            cout << "*";
13        }
14        cout << endl;
15    }
16
17    return 0;
18 }
```

	→ 4 個空白 1 個星星
	→ 3 個空白 3 個星星
	→ 2 個空白 5 個星星
	→ 1 個空白 7 個星星
	→ 0 個空白 9 個星星

→ 第 i 行 = $n-i$ 個空白 和 $i*2-1$ 個星星

利用迴圈印出圖形






```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++){
8         for (int j = 1; j <= n-i; j++){
9             cout << " ";
10        }
11        for (int j = 1; j <= i*2-1; j++){
12            cout << "*";
13        }
14        cout << endl;
15    }
16
17    return 0;
18 }
```

	→ 4 個空白 1 個星星
	→ 3 個空白 3 個星星
	→ 2 個空白 5 個星星
	→ 1 個空白 7 個星星
	→ 0 個空白 9 個星星

→ 第 i 行 = $n-i$ 個空白 和 $i*2-1$ 個星星

利用迴圈印出圖形

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7     for (int i = 1; i <= n; i++){
8         for (int j = 1; j <= n-i; j++){
9             cout << " ";
10        }
11        for (int j = 1; j <= i*2-1; j++){
12            cout << "*";
13        }
14        cout << endl;
15    }
16
17    return 0;
18 }
```

	→ 4 個空白 1 個星星
	→ 3 個空白 3 個星星
	→ 2 個空白 5 個星星
	→ 1 個空白 7 個星星
	→ 0 個空白 9 個星星

→ 第 i 行 = $n-i$ 個空白 和 $i*2-1$ 個星星

作業 #09

■#練習：請根據輸入的數值看印出對應高度的菱形圖形：

■Requirements：

1. 輸入一個正整數高度
2. 請輸出符合條件的圖形

■Sample Input：7

■Sample Output：參考右圖

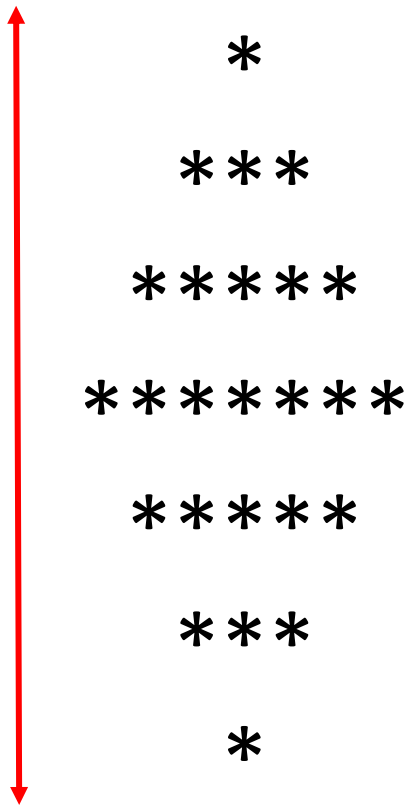
■Bonus：迴圈數量小於等於三個

■Hint：利用 `abs()` 取絕對值

■Note：僅限 10/20、11/03 上課繳交

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

高度有 7 行



高度有 7 行

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

→ 3 個空白 1 個星星

→ 2 個空白 3 個星星

→ 1 個空白 5 個星星

→ 0 個空白 7 個星星

→ 1 個空白 5 個星星

→ 2 個空白 3 個星星

→ 3 個空白 1 個星星

高度有 7 行

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

→ 3 個空白 1 個星星

→ 2 個空白 3 個星星

→ 1 個空白 5 個星星

→ 0 個空白 7 個星星

→ 1 個空白 5 個星星

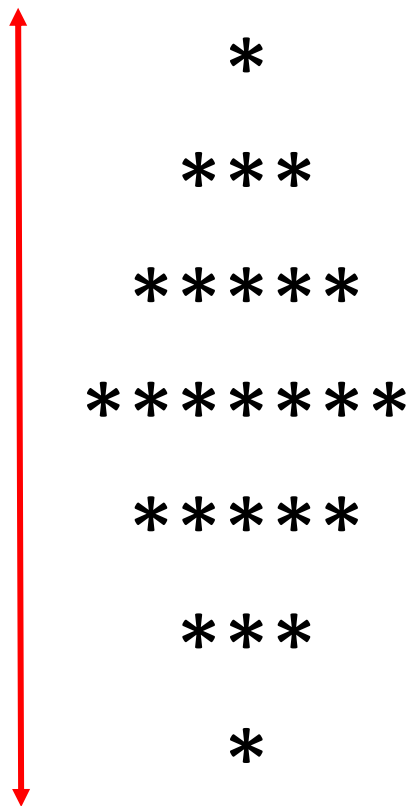
→ 2 個空白 3 個星星

→ 3 個空白 1 個星星

$i = 1, 2, 3, 4, u = 4$

$i = 1, 2, 3, d = 3$

高度有 7 行



高度有 7 行



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

第 1 行

→ 4 個空白 1 個星星

第 2 行

→ 3 個空白 3 個星星

第 3 行

→ 2 個空白 5 個星星

第 4 行

→ 1 個空白 7 個星星

第 5 行

→ 2 個空白 5 個星星

第 6 行

→ 3 個空白 3 個星星

第 7 行

→ 4 個空白 1 個星星

高度有 7 行



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

第 1 -3 行

→ 4 個空白 1 個星星

第 2 -2 行

→ 3 個空白 3 個星星

第 3 -1 行

→ 2 個空白 5 個星星

第 4 0 行

→ 1 個空白 7 個星星

第 5 1 行

→ 2 個空白 5 個星星

第 6 2 行

→ 3 個空白 3 個星星

第 7 3 行

→ 4 個空白 1 個星星

高度有 7 行



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

第 1 -3 行	→ 4 個空白 1 個星星
第 2 -2 行	→ 3 個空白 3 個星星
第 3 -1 行	→ 2 個空白 5 個星星
第 4 0 行	→ 1 個空白 7 個星星
第 5 1 行	→ 2 個空白 5 個星星
第 6 2 行	→ 3 個空白 3 個星星
第 7 3 行	→ 4 個空白 1 個星星

$i = -3, -2 \cdots 2, 3, n = 7 \ d = 3$

程式的邏輯思維

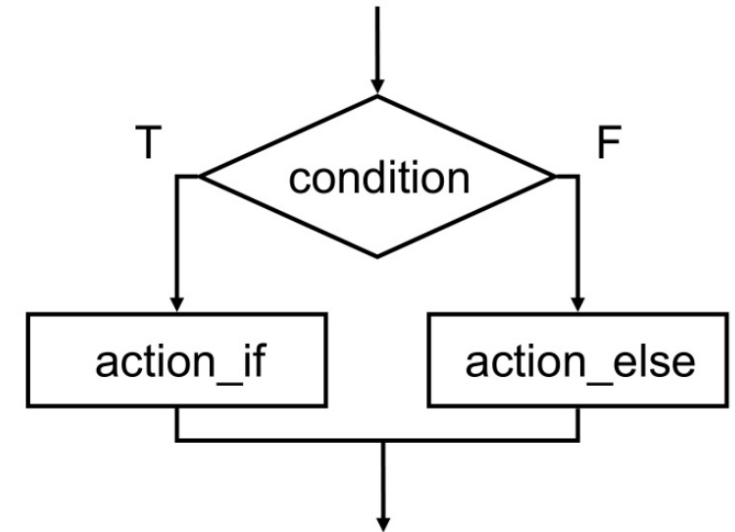
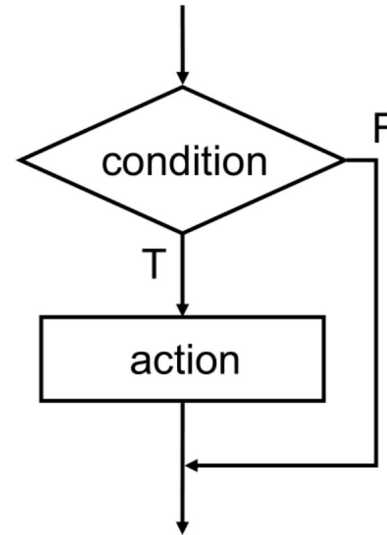
- 1. Sequential
- 2. Branch/Selection/Condition
 - if-else
- 3. Repeater/Iteration
 - for loop, while loop

程式的邏輯思維

- 1. Sequential : 由上而下、由右而左、由內而外
- 2. Branch/Selection/Condition
 - if-else
- 3. Repeater/Iteration
 - for loop, while loop

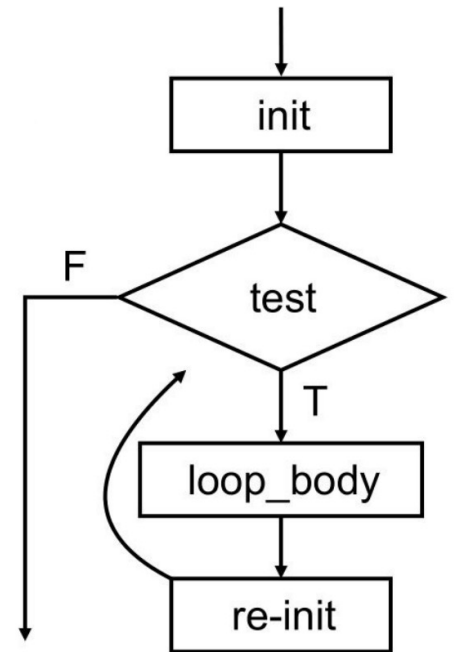
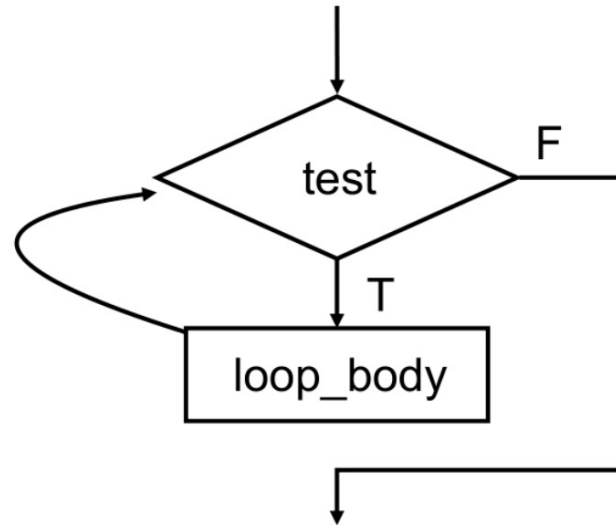
程式的邏輯思維

- 1. Sequential
- 2. Branch/Selection/Condition
 - if-else
- 3. Repeater/Iteration
 - for loop, while loop



程式的邏輯思維

- 1. Sequential
- 2. Branch/Selection/Condition
 - if-else
- 3. Repeater/Iteration
 - for loop, while loop



程式的邏輯思維

- 1. Sequential
- 2. Branch/Selection/Condition
 - if-else
 - switch
 - ?: (三元運算子)
- 3. Repeater/Iteration
 - for loop, while loop
 - do while

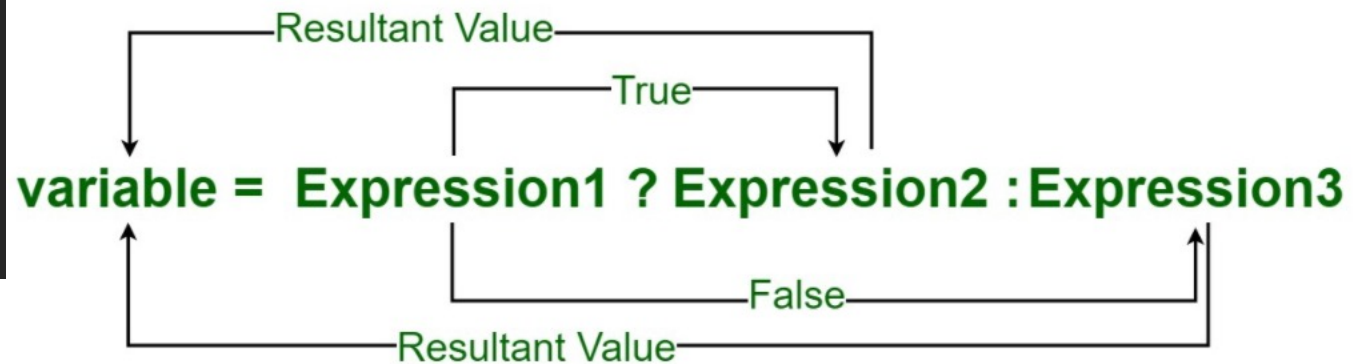
三元運算子

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int time = 20;
7
8     if (time < 18) {
9         cout << "Good day.";
10    } else {
11        cout << "Good evening.";
12    }
13
14    return 0;
15 }
16
17
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int time = 20;
7
8     string result = (time < 18) ?
9         "Good day." : "Good evening.";
10
11    cout << result;
12
13    return 0;
14 }
15
16
17
```

三元運算子

```
1
2 #include <iostream>
3 using namespace std;
4
5 int main(){
6
7     int time = 20;
8     string result = (time < 18) ? "Good day." : "Good evening.";
9
10    cout << result;
11
12    return 0;
13 }
14
15
16
17
```



幾種不同的條件寫法

```
if(x == 1){  
    y = 0  
} else {  
    y = 1  
}
```


幾種不同的條件寫法

```
if(x == 1){  
    y = 0  
} else {  
    y = 1  
}
```

```
if(x != 0){  
    y = 0  
} else {  
    y = 1  
}
```

幾種不同的條件寫法

```
if(x == 1){  
    y = 0  
} else {  
    y = 1  
}
```

```
if(x != 0){  
    y = 0  
} else {  
    y = 1  
}
```

```
if(x){  
    y = 0  
} else {  
    y = 1  
}
```

幾種不同的條件寫法

```
if(x == 1){  
    y = 0  
} else {  
    y = 1  
}
```

```
if(x != 0){  
    y = 0  
} else {  
    y = 1  
}
```

```
if(x){  
    y = 0  
} else {  
    y = 1  
}
```

```
y = 1  
if(x == 1){  
    y = 0  
}
```

```
y = 1  
if(x != 0){  
    y = 0  
}
```

```
y = 1  
if(x){  
    y = 0  
}
```

幾種不同的條件寫法

```
if(x == 1){  
    y = 0  
} else {  
    y = 1  
}
```

```
if(x != 0){  
    y = 0  
} else {  
    y = 1  
}
```

```
if(x){  
    y = 0  
} else {  
    y = 1  
}
```

```
y = 1  
if(x == 1){  
    y = 0  
}
```

```
y = 1  
if(x != 0){  
    y = 0  
}
```

```
y = 1  
if(x){  
    y = 0  
}
```

$y = x == 1 ? 0 : 1$

$y = x != 0 ? 0 : 1$

$y = x ? 0 : 1$

幾種不同的條件寫法

```
if(x == 1){  
    y = 0  
} else {  
    y = 1  
}
```

y = !x

Switch

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int day = 4;
6     switch (day) {
7         case 6:
8             cout << "Today is Saturday";
9             break;
10        case 7:
11            cout << "Today is Sunday";
12            break;
13        default:
14            cout << "Looking forward to the Weekend";
15    }
16    return 0;
17 }
```

do while

```
1
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6
7     int i = 0;
8     do {
9         cout << i << "\n";
10        i++;
11    } while (i < 5);
12
13    return 0;
14 }
15
```

利用迴圈輸入多筆測資

我們前面的都是依照著「循序 (sequential) 」的規則，從頭執行到尾，執行完結束就結束了… XD

→ 那如果想要有多次的輸入怎麼辦？

① 讀入 n 筆資料

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7     int i;
8
9     while (n-->0) {
10         cin >> i;
11         cout << "你輸入的是：" << i << endl;
12         cout << "-- 剩下 " << n << " 次輸入 --\n";
13     }
14
15     return 0;
16 }
```

① 讀入 n 筆資料

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7     int i;
8
9     while ( n != 0 ) {
10         n--;
11         cin >> i;
12         cout << "你輸入的是：" << i << endl;
13         cout << "-- 剩下 " << n << " 次輸入 --\n";
14     }
15
16     return 0;
17 }
```

① 讀入 n 筆資料

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7
8     for(int i = 1; i <= n; i++){
9         int N;
10        cin >> N;
11        cout << "你輸入的是：" << N << endl;
12    }
13    return 0;
14 }
15
16
```

② 讀到輸入結束

```
1
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7
8     int i;
9     while (cin >> i) {
10         cout << "使用者輸入:" << i << endl;
11     }
12
13     return 0;
14 }
15
```

Break and Continue

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7
8     for (int i = 1; i <= n; i++){
9         if(i == 3) break;
10        cout << "<= >" << i << endl;
11    }
12
13    return 0;
14 }
15
16
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     int n = 5;
7
8     for (int i = 1; i <= n; i++){
9         if(i == 3) continue;
10        cout << "<= >" << i << endl;
11    }
12
13    return 0;
14 }
15
16
```

Thanks for listening.

元智大學 | C++ 程式設計實習

Wei-Yuan Chang