1

## 目前作業或是隨堂練習出現的問題

- 資料型別的選擇
  - 如果資料或是運算過程可能會出現小數點 (e.g. BMI)，宜使用浮點數 (double, float)
  - char 僅能儲存一個英文字、數字、或是英文中出現的標點符號等等鍵盤上可直接看得到、按得出的符號
  - 若要儲存中文字，目前建議使用 string 型別。
  - 'A' 和 "A" 的差別，我們目前還沒辦法說得很詳細。只能說一個是字元、一個是字串。

2

## 目前作業或是隨堂練習出現的問題

- 布林運算式的寫法
  - 如果 x 等於 3，就印出 "Hello"
  - 如果 x 等於 3 或是 5，就印出 "Hello"
  - 如果 x 大於 3 且小於 8，就印出 "Hello"
  - 如果 x>=y >= z，就印出 "Hello"

```
if(x=3) cout << "Hello";
```

```
if(x==3 || 5) cout << "Hello";
```

```
if(x==3 < 8) cout << "Hello";
```

```
if(x>=y>=z) cout << "Hello";
```

3

## 目前作業或是隨堂練習出現的問題

- 布林運算式的正確寫法
  - 如果 x 等於 3，就印出 "Hello"
  - 如果 x 等於 3 或是 **x 等於** 5，就印出 "Hello"
  - 如果 x 大於 3 且 **x** 小於 8，就印出 "Hello"
  - 如果 x >=y **而且 y** >= z，就印出 "Hello"

```
if(x==3) cout << "Hello";
```

```
if(x==3 || x==5) cout << "Hello";
```

```
if(x>3 && x<8) cout << "Hello";
```

```
if(x>=y && y>=z) cout << "Hello";
```

4

## 目前作業或是隨堂練習出現的問題

- 一次輸入多筆資料

```
cin >> x, y, z;
```

```
cin >> x >> y >> z;
```

5

## 目前作業或是隨堂練習出現的問題

- 平方
  - 3 的平方

```
#include<iostream>
using namespace std;

int main() {
    int x = 3;
    int y = x ^ 2;

    cout << y << endl;

    return 0;
}
```

^ 的確是 C/C++ 裡的一個運算子，但是它的運算不是大家在 Excel 裡知道的運算。

^ 進行的運算稱為 「exclude or」，簡寫為 xor。

```
1
```

6

## 回顧

for (____; ____; ____ ) {
}

break;

continue;

7

## for 迴圈

```
    1   2   3
for (i=0; i<3; i++) {
4           cout << “a”;
}

5 cout << “b”;
```

1 迴圈被執行到時初始化的動作
2 迴圈要執行或要被繼續執行的條件
3 迴圈完成一次後要做的動作
4 迴圈內欲重複執行的敘述
5 迴圈結束後要執行的敘述

| | |
|---|---|
| 1 | 令 i = 0 |
| 2 | 0 < 3 → *true* |
| 4 | cout << “a”; |
| 3 | i++; |
| 2 | 1 < 3 → *true* |
| 4 | cout << “a”; |
| 3 | i++; |
| 2 | 2 < 3 → *true* |
| 4 | cout << “a”; |
| 3 | i++; |
| 2 | 3 < 3 → ***false*** |
| 5 | cout << “b”; |

8

## 6-1.cpp

```
int x, i;
cout << "\n請輸入一個正整數; ";
cin >> x;

for(i=2;i<=x;i++) {
      if(x % i == 0) break;
}

if(i == x)
      cout << x << " 為質數。";
else
      cout << x << " 不是質數，可以被" << i << " 整除。";
```

9

# Lecture 6

while
do ... while
無窮迴圈
巢狀迴圈

10

## 今日內容

- while 迴圈
- do/while 迴圈
- 巢狀迴圈
- 無窮迴圈

11

# while 迴圈

12

## while 的語法

- while (條件運算式) {
  欲重複執行的程式片斷;
  ...
  }
- 其中條件運算式 / 布林運算式的部份與 if 的寫法一樣
  - count>0
  - a<10 && b<10
  - ...
- while 和 for 迴圈一樣，都會**先檢查**條件運算式的結果決定迴圈的內容是否會被執行，所以**有可能一次都不會被執行**。

13

## 6-2.cpp

```
請輸入一個整數: 10
count=10, sum=10
count=9, sum=19
count=8, sum=27
count=7, sum=34
count=6, sum=40
count=5, sum=45
count=4, sum=49
count=3, sum=52
count=2, sum=54
count=1, sum=55
```

```
#include <iostream>
using namespace std;
int main() {
  int count, sum=0;

  cout << "請輸入一個整數: ";
  cin >> count;
  while(count>0) {
      sum = sum + count;
      cout<< "count=" << count << ", sum=" << sum << endl;
      count--;
  }

  return 0;
}
```

14

## 6-3.cpp

```
請輸入一個偶數: 1
請輸入一個偶數: 3
請輸入一個偶數: 5
請輸入一個偶數: 7
請輸入一個偶數: 6
你輸入了一個偶數: 6
```

```
#include <iostream>
using namespace std;
int main() {
  int ans;
  bool isOdd = true;

  while(isOdd) {
      cout << "請輸入一個偶數: ";
      cin >> ans;

      isOdd = (ans % 2 == 1);
  }
  cout << "你輸入了一個偶數: " << ans << endl;

  return 0;
}
```

15

## 6-4.cpp

- 費氏級數 (Fibonacci number) 之定義如下:
  - 1, 1, 2, 3, 5, 8, 13, 21, ...
  - 級數中每個數字皆為級數中的前兩個數字加總
- 如何寫一個程式依序列出級數中小於 150 的各項?
  1. 我們自己會怎麼作?
  2. 過程當中我們需要同時記得幾個資料?

16

## 6-4.cpp

```
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,
```

```
#include <iostream>
using namespace std;
int main() {
  int a=1, b=1, c=a+b;

  cout << a << ", " << b << ", ";
  while(c<=150) {
      a=b;
      b=c;
      cout << b << ", ";
      c=a+b;
  }

  return 0;
}
```

17

## do/while 迴圈

18

## do/while 的語法

- do {
  欲重複執行的程式片段;
  ...
  } while (布林運算式);  很重要
- 其中布林運算式的部份與 if 的寫法一樣
  - count>0
  - a<10 && b<10
  - ...
- do/while 迴圈與 while 或 for 最大不一樣的地方在於它的條件判斷在迴圈內容之後，亦即它的迴圈內容**被保證至少執行到一次**。
- 記得 while ( ... ) 之後要加分號！

19

## 6-5.cpp

```
#include <iostream>
using namespace std;
int main() {
  char again;
  float weight, height, BMI;
  do {
      cout << "請輸入你的身高 (公分): ";
      cin >> height;
      cout << "請輸入你的體重 (公斤): ";
      cin >> weight;
      BMI = weight*10000 / height / height;
      cout << "你的 BMI 值為: " << BMI << endl;
      cout << "再算一次請按 y: ";
      cin >> again;
  } while(again=='y' || again=='Y');
  return 0;
}
```

```
請輸入你的身高 (公分): 170
請輸入你的體重 (公斤): 60
你的 BMI 值為: 20.7612
再算一次請按 y: y
請輸入你的身高 (公分): 180
請輸入你的體重 (公斤): 60
你的 BMI 值為: 18.5185
再算一次請按 y: y
請輸入你的身高 (公分): 172
請輸入你的體重 (公斤): 55
你的 BMI 值為: 18.5911
再算一次請按 y: n
```

20

## 6-6.cpp

```
#include <iostream>
using namespace std;
int main() {
  int ans;

  do {
      cout << "請輸入一個偶數: ";
      cin >> ans;
  } while(ans%2 != 0);

  cout << "你輸入了一個偶數" << endl;

  return 0;
}
```

```
請輸入一個偶數: 1
請輸入一個偶數: 3
請輸入一個偶數: 5
請輸入一個偶數: 7
請輸入一個偶數: 6
你輸入了一個偶數
```

21

## 小結

- while
- do ... while
- 注意, 之前介紹過的 break 與 continue 一樣也可以使用在 while / do-while 迴圈內

22

## while 迴圈

```
int a = 0;
1
while( a < 3) {
2       cout << "a";
        a++;
}
3 cout << "b";
```

1 判斷迴圈內容是否執行或繼續執行
2 迴圈內欲重複執行的敘述
3 迴圈結束後要執行的敘述

1 0 < 3 → *true*
2 cout << "a";
2 a++;
1 1 < 3 → *true*
2 cout << "a";
2 a++;
1 2 < 3 → *true*
2 cout << "a";
2 a++;
1 2. 3 < 3 → ***false***
3 4. cout << "b";

23

## do/while 迴圈

```
int a = 0;

do {
1       cout << "a";
        a++;
2 } while(a<3);
3 cout << "b";
```

1 迴圈內欲重複執行的敘述
2 迴圈要執行或要被繼續執行的條件
3 迴圈結束後要執行的敘述

1 cout << "a";
1 a++;
2 1 < 3 → *true*
1 cout << "a";
1 a++;
2 2 < 3 → *true*
1 cout << "a";
1 a++;
2 3 < 3 → ***false***
3 cout << "b";

24

## 無窮迴圈

## Infinite loop

25

## 無窮迴圈 (Infinite loop)

- 沒有結束的迴圈
  - 用於無論在迭代開始 (for, while) 或是結束 (do/while) 來判斷是否進行下次迭代都不合適的時候，一般配合 break 使用。
  - 真的不想結束程式的時候 ...

- `for(;;) {…}`
- `while(true) {…}`
- `do {…} while(1);`

26

## 6-7.cpp

這個 else 有沒有差別？

對程式執行結果而言，沒有！
就程式效率而言，有！

```
int choice;

for(;;) {
  cout << "\n歡迎使用本程式";
  cout << "\n輸入 1) 問候, 2) 感謝, 3) 再見: ";
  cin >> choice;
  if(choice==3) break;
  if(choice==1) {
    cout << "\nHello there.  How are you? ";
  }
  else if(choice==2) {
    cout << "\nThank you ver much. ";
  }
}

cout << "\nSee you next time" << endl;
```

27

```
歡迎使用本程式
輸入 1) 問候, 2) 感謝, 3) 再見: 1

Hello there.  How are you?
歡迎使用本程式
輸入 1) 問候, 2) 感謝, 3) 再見: 2

Thank you ver much.
歡迎使用本程式
輸入 1) 問候, 2) 感謝, 3) 再見: 1

Hello there.  How are you?
歡迎使用本程式
輸入 1) 問候, 2) 感謝, 3) 再見: 3

See you next time
```

28

## 巢狀迴圈

## nested loop

29

## 巢狀迴圈

- 就是迴圈裡面還有迴圈
  - for, while, do/while 三種迴圈可以互相混合
- 層數幾乎沒有限制
- 被包起來的迴圈稱作「內迴圈」(inner-loop)，包其它迴圈的稱作「外迴圈」(outer-loop)。
  - 外迴圈 (i) 通常不會使用或參考到內迴圈的計數器 (j)
  - 內外層的迴圈計數器的增量**通常**獨立控制
  - 內迴圈 (j) 有時會參考到外迴圈 (i) 的計數器

第一層迴圈 / 第二層迴圈

```
for(i=0;i<3;i++) {
  cout << "*, i=" << i << endl;
  for(j=0;j<4;j++) {
    cout << "**, i=" << i << ", j=" << j << endl;
  }
}
```

30

## 6-8.cpp

```
#include <iostream>
using namespace std;

int main() {
  int j;

  for(int i=10; i<=20; i=i+5) {
    cout << "* " << i << endl;
    for(j=2; j<=4; j++) {
      cout << "** " << i << ", " << j << endl;
    }
    cout << "+ " << i << ", " << j << endl;
  }
  return 0;
}
```

```
* 10
** 10, 2
** 10, 3
** 10, 4
+ 10, 5
* 15
** 15, 2
** 15, 3
** 15, 4
+ 15, 5
* 20
** 20, 2
** 20, 3
** 20, 4
+ 20, 5
```

31

## 6-8.cpp

```
int j;
                1a      1b      1c
    for(int i=10; i<=20; i=i+5) {
2     cout << "* " << i << endl;
          3a    3b    3c
      for(j=2; j<=4; j++) {
4       cout<<"** "<<i<<", "<<j<<endl;
      }
5     cout<<"+ "<<i<<", "<<j<<endl;
    }
6
```

```
1a 1b  2   * 10
3a 3b  4   ** 10, 2
3c 3b  4   ** 10, 3
3c 3b  4   ** 10, 4
3c 3b  5   + 10, 5
1c 1b  2   * 15
3a 3b  4   ** 15, 2
3c 3b  4   ** 15, 3
3c 3b  4   ** 15, 4
3c 3b  5   + 15, 5
1c 1b  2   * 20
3a 3b  4   ** 20, 2
3c 3b  4   ** 20, 3
3c 3b  4   ** 20, 4
3c 3b  5   + 20, 5
1c 1b  6
```

32

## 6-9.cpp

```
#include <iostream>
using namespace std;

int main() {
  int i, j;

  for(i=0; i<5; i++) {
    for(j=0; j<4; j++) {
      cout << "*";
    }
    cout << endl;
  }

  return 0;
}
```

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

33

## 6-10.cpp

```
#include <iostream>
using namespace std;

int main() {
  int i, j;

  for(i=0; i<10; i++) {
    for(j=0; j<i; j++) {
      cout << "*";
    }
    cout << endl;
  }

  return 0;
}
```

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

34

## 6-11.cpp

- 我們想列出兩位數字中 (總共有幾個?) 各個位數不相同者，並計算總共有幾個符合條件的數字。
  - e.g. 11, 22, 33 等不符合條件
  1. 我們怎麼列出所有兩位數?
  2. 我們怎麼將需求之條件 (各個位數不相同者) 寫成程式?

35

## 6-11.cpp

```
int count=0;

for(int i=1;i<10;i++) {
  for(int j=0;j<10;j++) {
    if( i != j ) {
      cout << i << j << endl;
      count++;
    }
  }
}
cout << "共有: " << count << endl;
```

36

## 隨堂練習

37

## 1. 排列組合

- 請寫一程式，列出 0 – 4 所組成的三位數中互相不重複的所有數字，並印出共有幾個符合條件的數字。
  - Hint 1: 每一位數由一個迴圈控制
  - Hint 2: 使用條件判斷 + continue 或是 break 跳過不要的數字
  - Hint 3: 如果自己想不出來的話，可以參考 6-11.cpp

38

## 參考輸出

```
102     201     301     401
103     203     302     402
104     204     304     403
120     210     310     410
123     213     312     412
124     214     314     413
130     230     320     420
132     231     321     421
134     234     324     423
140     240     340     430
142     241     341     431
143     243     342     432
```