

# C++程式設計基礎

## lesson 2

陳毅

# 上週回顧

- 基本輸入輸出
- 資料型態
- 宣告變數與常數、指定資料

# 基本輸入輸出

- 輸入函數：`cin`
  - 配合「`>>`」使用
  - 讀取輸入直到空格或換行符
- 輸出函數：`cout`
  - 配合「`<<`」使用
  - 換行符：`endl`, “`\n`”

# 資料型態

- 整數
  - int, short, long
- 浮點數（小數）
  - float, double
- 字元
  - char（範圍0~255）
  - ASCII碼表：<https://zh.wikipedia.org/wiki/ASCII>
- 布林值
  - bool（邏輯運算符）
  - 真或假、true or false

# 宣告變數與常數、指定資料

**資料型態 變數名稱1, 變數名稱2, ...;**

- 變數名稱的命名規則
  - 由英文字母、數字、底線符號組合而成，開頭不可為數字。
  - 會區分大小寫。
  - 注意：變數名稱的命名與變數的儲存內容完全無關。

# 練習

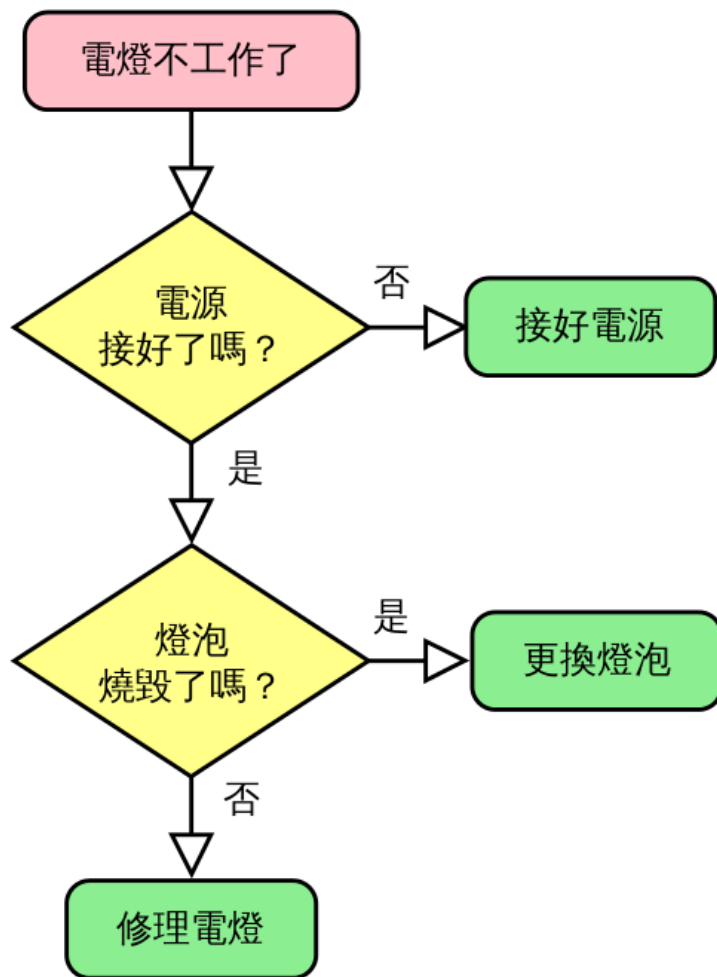
- 請寫一個程式，輸入一個人的身高及體重，輸出該人的 **BMI**。

# 本週概要

- 程式結構
  - 循序式結構 (sequence structure)
  - 選擇式結構 (selection structure)
  - 重複式結構 (repetition structure)
- 條件控制
  - if, if-else, if-else if, switch
- 迴圈控制
  - for, while, do-while

# 程式結構

- 流程圖





# 程式結構

- 循序式結構
  - 從第一個敘述執行至最後一個敘述。
- 選擇式結構
  - 程式中有條件敘述，當 **條件成立** 時，執行 條件成立區 的敘述。反之，**條件不成立** 時，執行 條件不成立區 的敘述。
- 重複式結構
  - 程式中有重複敘述，當 **重複敘述的條件成立** 時，執行 重複執行區 的敘述。

條件控制

# 條件控制

```
a1 = 3;  
a2 = 4;
```

```
bool b = a1 > a2; // b = false;
```

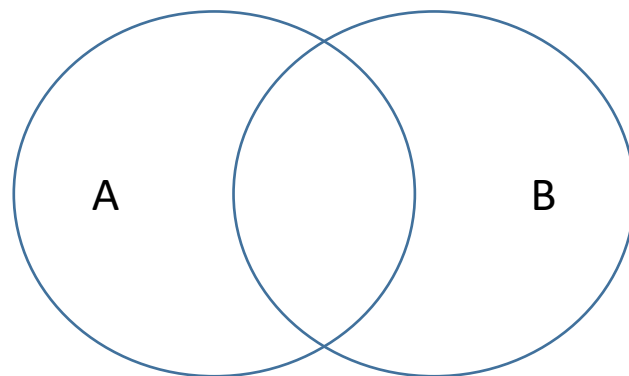
- 關係運算符

符號	功能	範例	說明
>	大於	<code>a &gt; b</code>	若a大於b，則結果為真。
<	小於	<code>a &lt; b</code>	若a小於b，則結果為真。
>=	大於等於	<code>a &gt;= b</code>	若a大於等於b，則結果為真。
<=	小於等於	<code>a &lt;= b</code>	若a小於等於b，則結果為真。
==	等於	<code>a == b</code>	若a等於b，則結果為真。
!=	不等於	<code>a != b</code>	若a不等於b，則結果為真。

! not

# 條件控制

- 邏輯運算符



&&

true	true	true
true	false	false
false	true	false
false	false	false

符號	功能	範例	說明
!	NOT	!(a==1)	若a不等於1，則結果為真
&&	AND	a > 1 && a < 9	若1<a<9，則結果為真
	OR	a < 1    a > 9	若a<1或a>9，則結果為真

- 範例一：驗證一個數是否為2、3、5的倍數。
- 範例二：驗證一個年份是否為閏年。
  - <https://zh.wikipedia.org/wiki/%E9%97%B0%E5%B9%B4>

# 條件控制

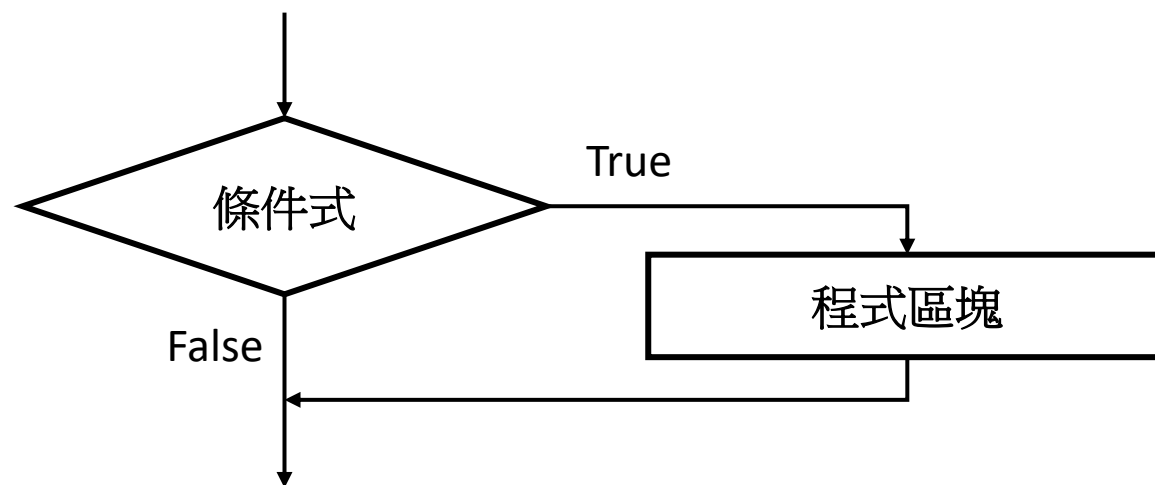
- if
- if...else...
- if...else if...else...
- switch

# 條件控制

- if敘述的用途，是判斷是否進入程式區塊，使用語法如下：

```
if (條件式){  
    程式區塊;  
    ...  
}
```

- 流程圖表示為



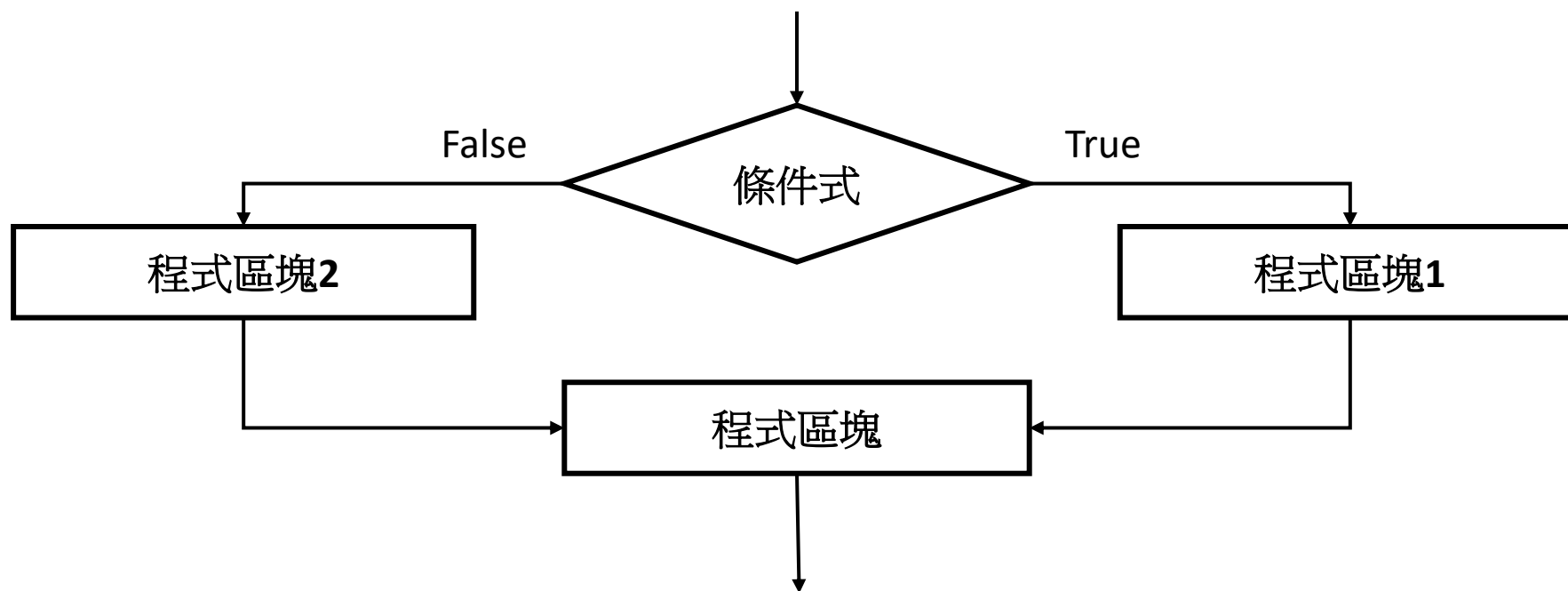
# 條件控制

- **if...else...**敘述會根據條件式的真偽，決定要進入的程式區塊為何。如果條件式成立，則會執行程式區塊**1**；否則會執行程式區塊**2**。使用語法如下：

```
if (條件式){  
    程式區塊1;  
    ...  
} else {  
    程式區塊2;  
    ...  
}
```

# 條件控制

- 流程圖表示為





# 條件控制

- if...else if...else...

```
if (條件式1){  
    程式區塊1;  
    ...  
} else if (條件式2){  
    程式區塊2;  
    ...  
    ...  
} else if (條件式N){  
    程式區塊N;  
    ...  
} else {  
    程式區塊N+1;  
}  
}
```

# 條件控制

- switch

- 注意

switch內的程式敘述為循序式結構，因此要在每個case後面加上break。

```
switch (變數/運算式){  
    case 值1:  
        程式敘述1;  
        ...  
        break;  
    case 值2:  
        程式敘述2;  
        ...  
        break;  
    ...  
  
    case 值N:  
        程式敘述N;  
        ...  
        break;  
    default:  
        程式敘述N+1;  
        ...  
        break;  
}
```

# 條件控制

- 百貨公司週年慶打折程式

豪慷慨百貨公司週年慶，公司決定在公司消費超過**2000**元的顧客就打**7**折，來增加買氣，請幫該公司寫出一個收銀台程式，輸入顧客購買總金額後，計算顧客實際需付的錢。

- 單位轉換程式

**1公尺=3.28**英呎，**1公斤=2.2**英鎊，請寫出一個可讓使用者自由選擇要轉換哪一單位的程式。

# 迴圈控制

# 迴圈控制

- 題目一

寫一個程式，計算 $1+2+3+\dots+100$ 的值

- 題目二

寫一個程式，列出1到1000間，3、5、7的倍數。

- 題目三

寫一個程式，設計一個終極密碼的遊戲。

1. 輸入通關密碼，數字必須在1~100之間。
2. 輸入數字後按下Enter鍵。
3. 若猜錯，就顯示縮小後的數字範圍，並依提示再次猜謎。
4. 若猜對，則顯示評語。

# 迴圈控制

- 什麼時候會用到迴圈 (loop)？

Ans: 要重複執行某一個敘述區的敘述時。

起始式：一般用於初始化「計數用的變數」

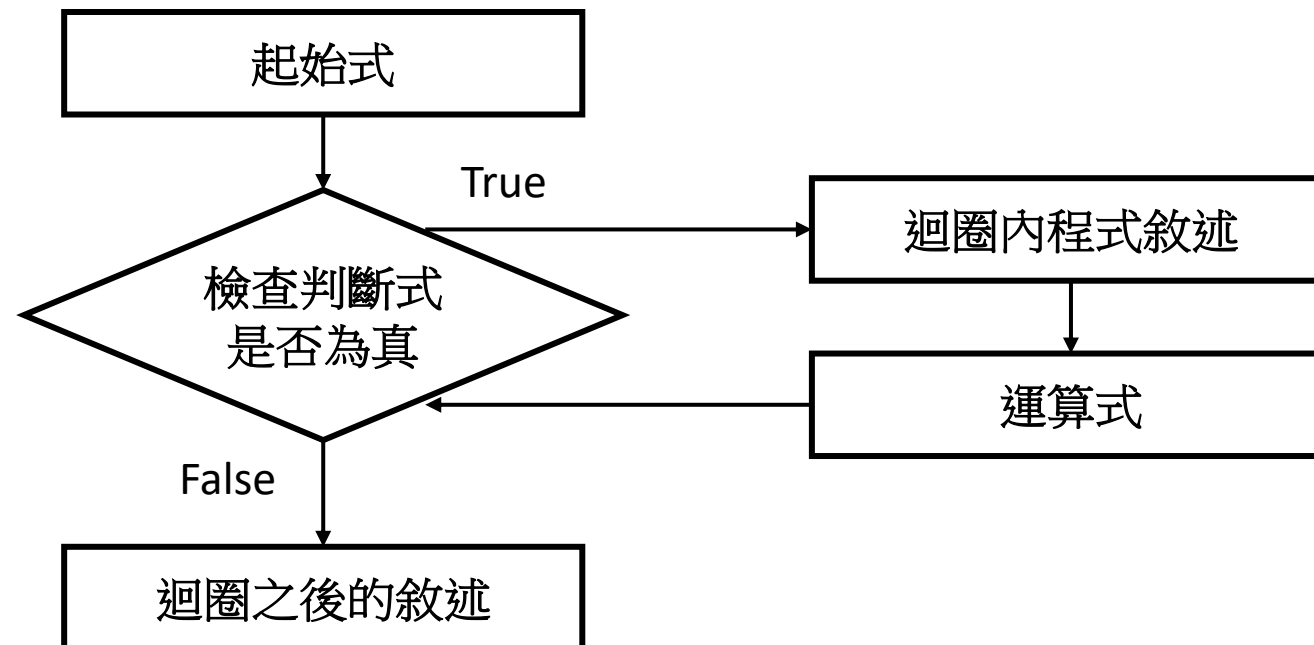
# 迴圈控制

for loop 的強項：用來執行有限次數的重複式結構

- for迴圈語法如右：

```
for (起始式; 判斷式; 運算式){  
    程式敘述;  
    ...  
}
```

- 流程圖：

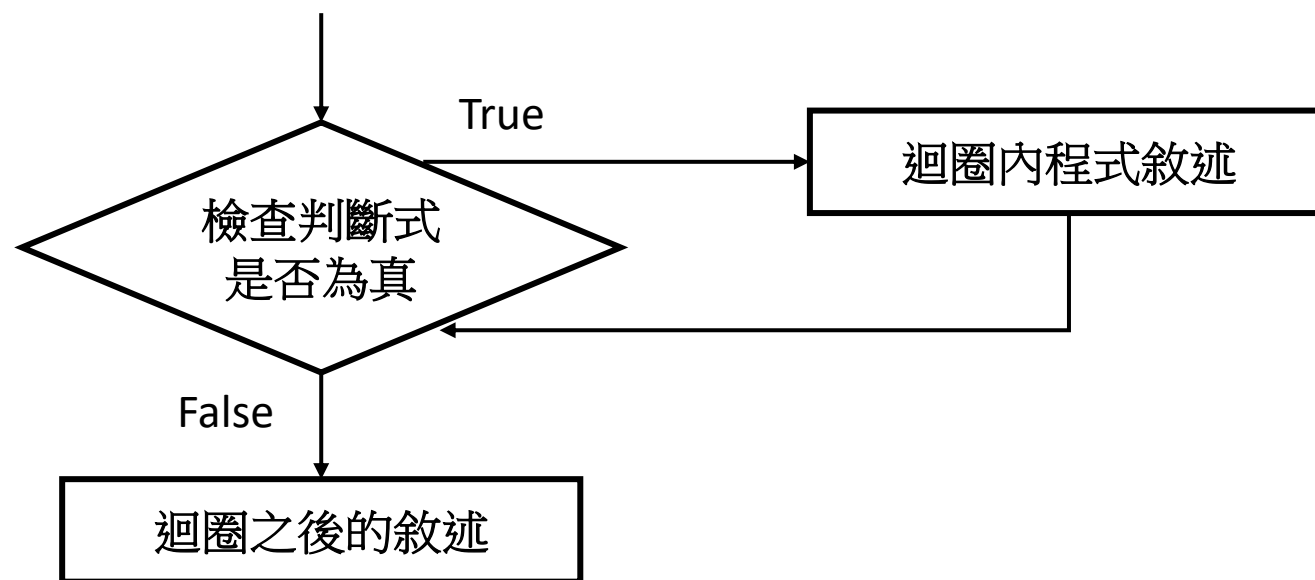


# 迴圈控制

- while迴圈與for迴圈相似，但是沒有起始式與運算式。

```
while (判斷式){  
    程式敘述;  
    ...  
}
```

- 流程圖：



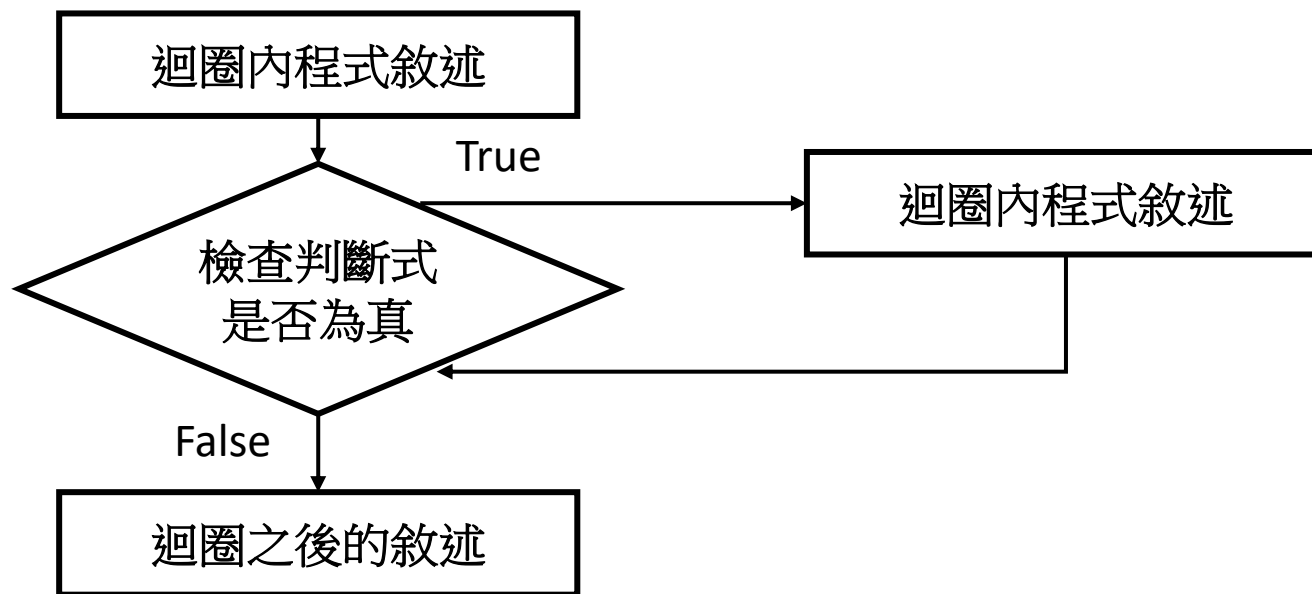


# 迴圈控制

- do...while迴圈是while迴圈的變形，差別在於迴圈內的程式敘述會先執行一次。

```
do {  
    程式敘述;  
    ...  
} while (判斷式);
```

- 流程圖：



# 迴圈控制

- **continue**敘述

- 迴圈可以使同一段程式敘述執行多次，但可能在某一些情形下，我們希望能在執行到某一敘述時，直接跳回迴圈的起點。
- 當程式執行到**continue**敘述時，會略過其下方的敘述，直接跳回迴圈起點。

- **break**敘述

- 當程式執行到**break**敘述時，會直接跳出迴圈，不再執行迴圈內的敘述。

# 迴圈控制

- 題目一

寫一個程式，計算 $1+2+3+\dots+100$ 的值

- 題目二

寫一個程式，列出1到1000間，3、5、7的倍數。

- 題目三

寫一個程式，設計一個終極密碼的遊戲。

1. 輸入通關密碼，數字必須在1~100之間。
2. 輸入數字後按下Enter鍵。
3. 若猜錯，就顯示縮小後的數字範圍，並依提示再次猜謎。
4. 若猜對，則顯示評語。

# 練習

- 條件控制
  - a020：身分證驗證
  - d584：技能點數skill
- 迴圈控制
  - a005：Eva的回家作業
  - a244：新手訓練 ~ for + if
  - d649：數字三角形
  - a022：迴文
- Reading: 課本Ch4~Ch5
- 若遇到作業問題，歡迎隨時寄信至：[ck1001099@gmail.com](mailto:ck1001099@gmail.com)