

C++

雙重選擇



C / C++ 語言提供三種選擇結構

選擇結構	描述
if選擇敘述式	單一選擇敘述式，只選擇執行或跳過一項動作
if...else選擇敘述式	雙重選擇敘述式，會取兩種不同動作之一
switch	依運算式的不同，選擇執行許多動作之一

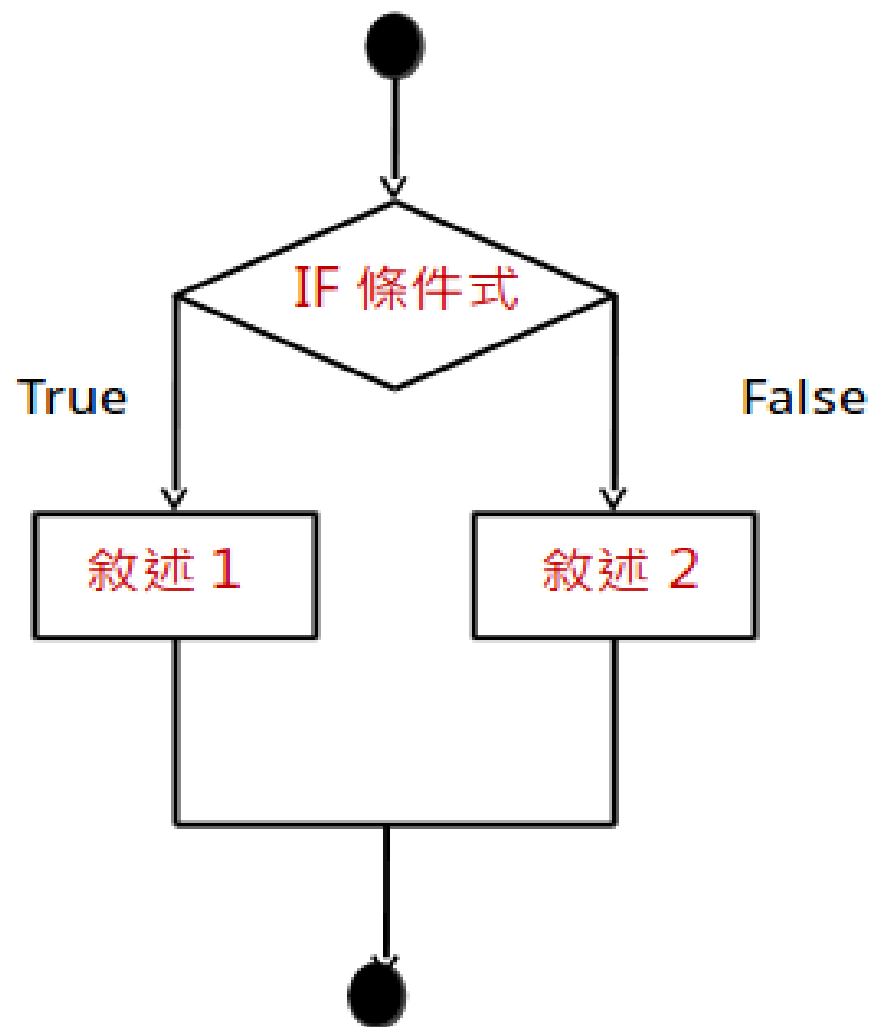
本單元主要學習if...else選擇敘述式



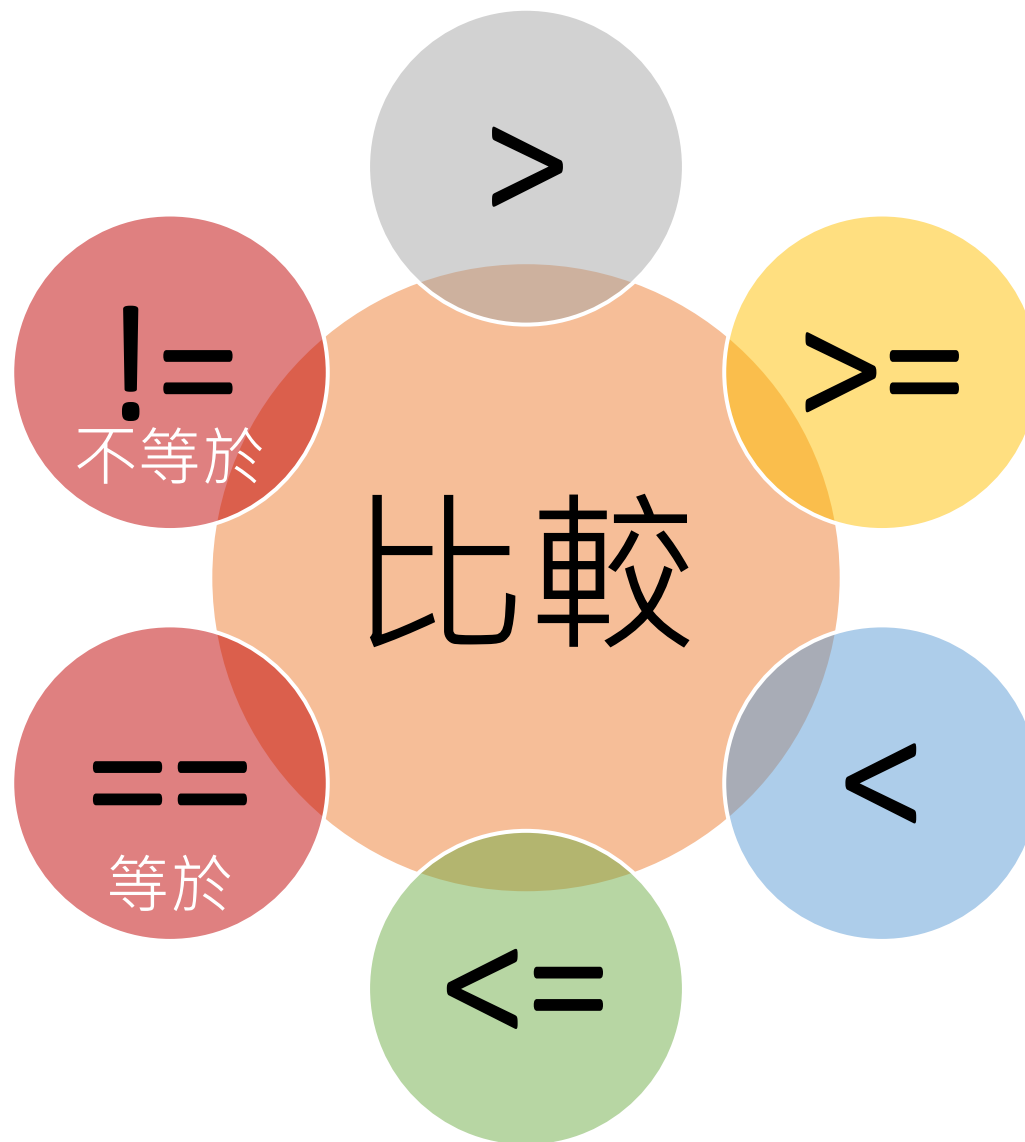
不是這樣就那樣

if...else

雙重選擇流程圖



關係運算子



相等運算子是==與指定運算子=常發生混淆

```
if (a==4)
    cout<<"You get a bonus!"<<endl;
```

但不小心寫成：

```
if (a=4)
    cout<<"You get a bonus!"<<endl;
```

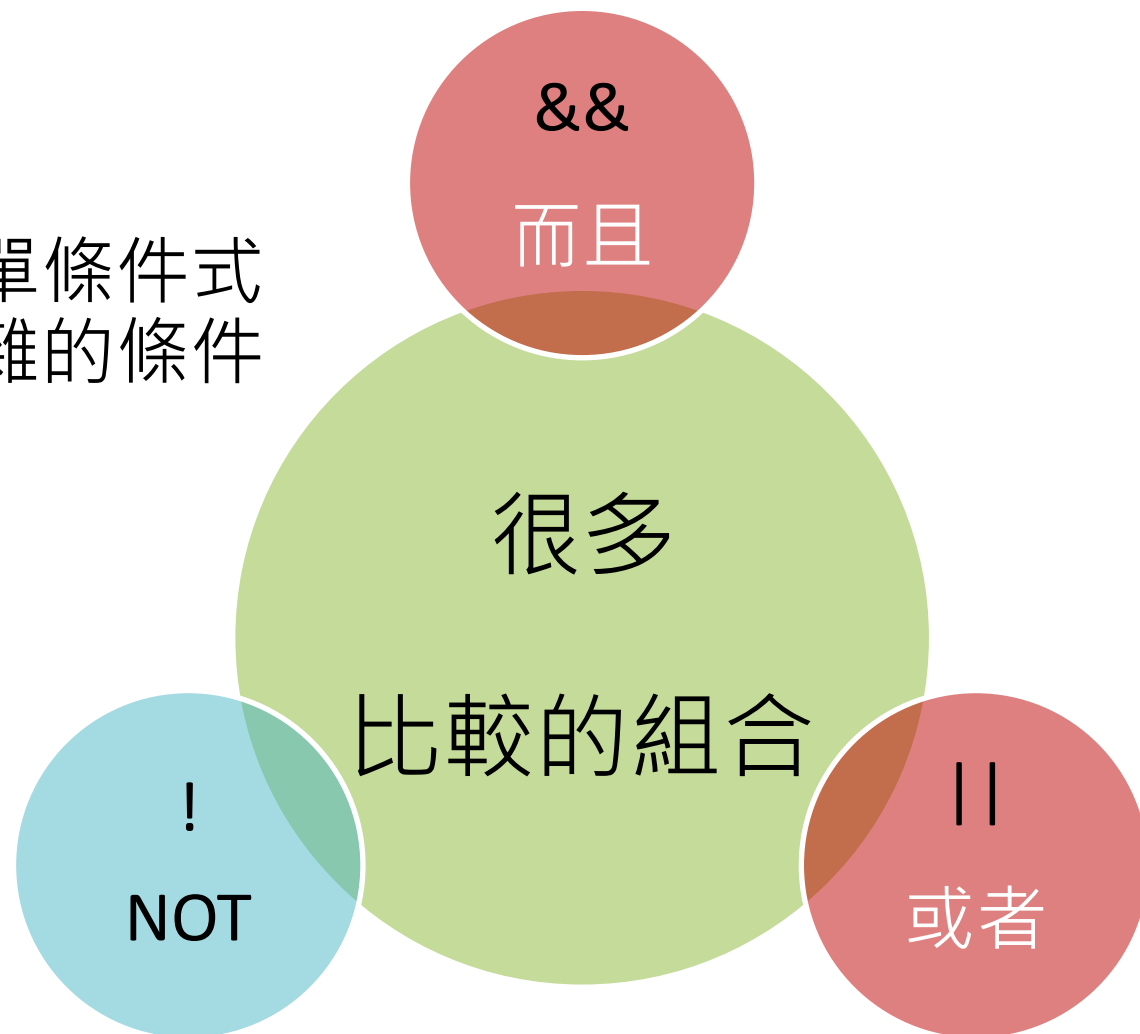


關係運算子與程式碼

代數符號	C語言符號	C的條件式範例	C條件式的意義
\neq	<code>!=</code>	<code>x!=y</code>	x不等於y
$=$	<code>==</code>	<code>x==y</code>	x等於y
$>$	<code>></code>	<code>x>y</code>	x大於y
$<$	<code><</code>	<code>x<y</code>	x小於y
\geq	<code>>=</code>	<code>x>=y</code>	x大於等於y
\leq	<code><=</code>	<code>x<=y</code>	x小於等於y

邏輯運算子

- 可以將數個簡單條件式組合成一個複雜的條件式



邏輯運算子

而且

運算符號	意涵	範例	說明	一元或二元運算
&&	and 運算	a && b	a與b做 and 邏輯運算	二元
	or運算	a b	a與b做 or 邏輯運算	二元
^	xor運算	a ^ b	a與b做 xor 邏輯運算	二元
!	not運算	!a	a做 not 邏輯運算	一元

xor

如果a、b兩個值不相同，則xor結果為1。
如果a、b兩個值相同，xor結果為0。

邏輯運算子-真值表

- 下表列出了運算式1與運算式2之0(false)與非0(true)值的四種可能組合，這個表稱真值表
- C會對所有的運算式(關係運算子與邏輯運算子)化簡成0或1。雖然C將真值設為1，不過只要是不為0的值都可接受當成真

運算式1	運算式2	a && b	a b	a xor b	not a
0	0	0	0	0	1
0	非0	0	1	1	
非0	0	0	1	1	0
非0	非0	1	1	0	

練習

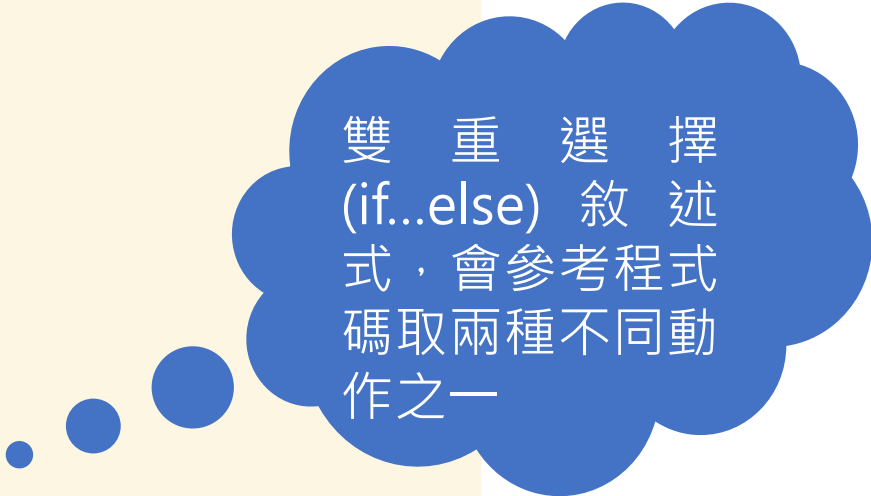
寫一個程式，判斷一個分數是否及格(60分以上)？



參考程式碼

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

int main(){
    int score;
    cin>>score;
    if(score>=60)
        cout<<"及格"<<endl;
    else
        cout<<"不及格"<<endl;
    return 0;
}
```



雙重選擇
(if..else) 敘述
式，會參考程式
碼取兩種不同動
作之一



C ++

延伸的概念

Extended concept



使用三元運算子表達 if...else

條件運算子是C/C++中唯一的三元運算子-它使用三個運算元

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

int main() {
    int score;
    cin>>score;
    cout << ((score>=60)?"及格":"不及格") << endl;
    return 0;
}
```

- 第一個運算元是條件
- 第二個運算元是當條件為真實，整個條件運算式的值
- 第三個運算元是當條件為假時，整個條件運算式的值

%s代表輸出的是字串，及格或不及格

再談寫作風格

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

int main(){
    int num;
    cin>>num;
    if(num%2==0)
        cout<<num<<"是偶數"<<endl;
    else
        cout<<num<<"是奇數"<<endl;
    return 0;
}
```

- If 與 else 的下方分別包含一個區塊，區塊是由大跨號 { } 所包圍，if 與 else 下方如果只包含一個敘述式，大跨號 { } 是可以被省略的，雖然 { } 被省略，if 與 else 下方還是被視為一個區塊。
- 第 8 行與第 10 行有縮排，程序的可讀性還是高一點。

學會將問題縮小範圍，逐步找出錯誤

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

int main() {
    int x,a,b,c;
    scanf("%d",&x);
    a = x / 100;

    if(a % 3 == 0)
        cout<<"百位數"<<a<<"是三的倍數\n";
    else
        cout<<"百位數"<<a<<"不是三的倍數\n";

    //if
        //printf
    //else
        //printf
    //if
        //printf
    //else
        //printf

    return 0;
}
```

- 這個題目涵蓋3個項目，一旦程序有問題，可以將答案拆成3個部分，一一的去找錯誤，錯誤很快就會被找出來，一旦錯誤被找出來，拿一支筆與一張紙，使用紙筆做一次演算，問題很快會被找到。
- 分段檢視程序代碼的方法，可以**試試//的功能，加上 //**，後面的程序代碼會被編譯程序忽略的。

區塊如果包含2行以上的敘述式，要加上{ }

```
if( 條件一 || 條件二 ){  
    cout<<"符合標準"<<endl;  
    cout<<n<<endl;  
}
```