

C++

單一選擇

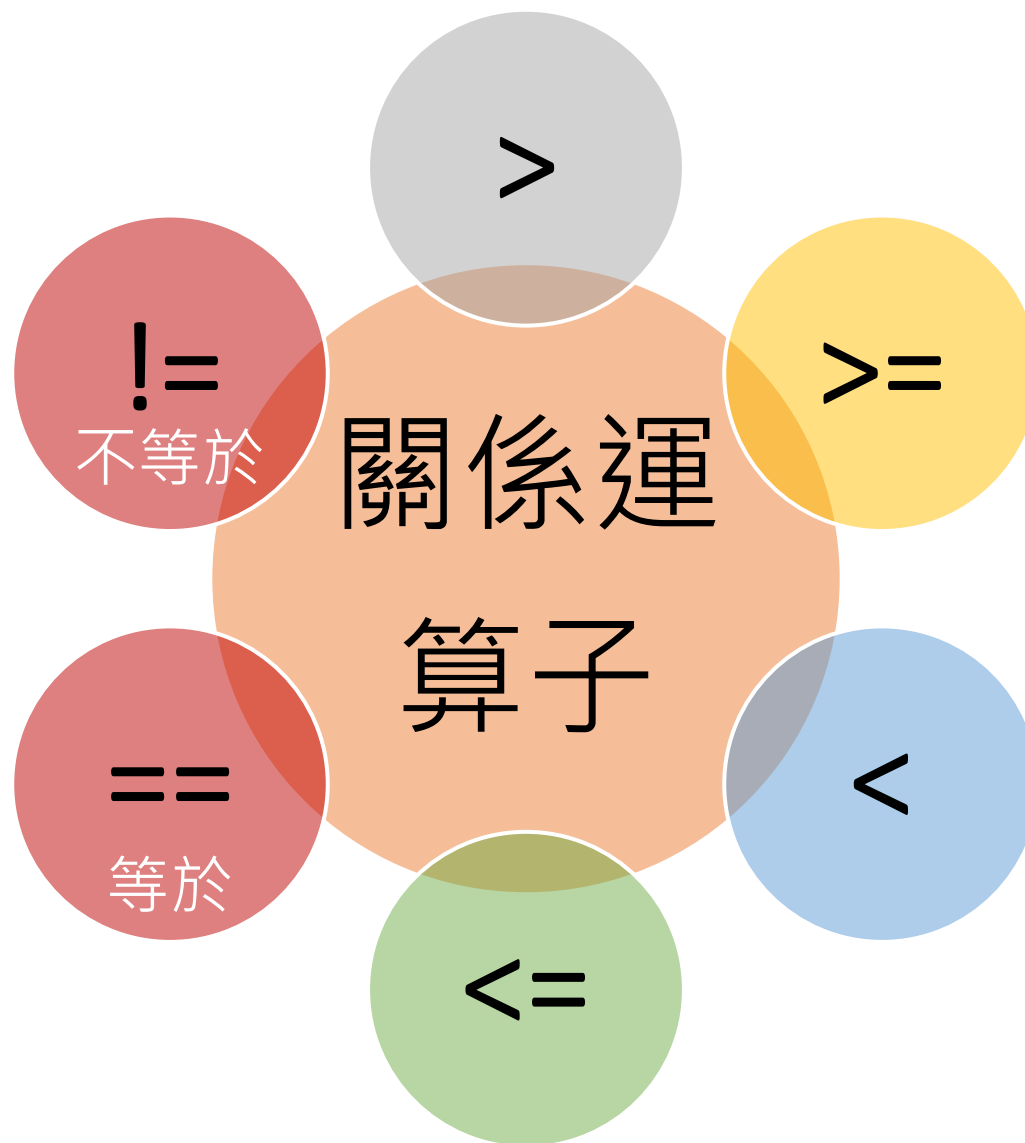


C / C++ 語言提供三種選擇結構

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選擇敘述式

選擇之前，需要比較



關係運算子(relational operator)

或是 比較運算子(comparison operator) (兩者相同)

關係/比較運算子	運算式	運算子的意義
==	$x==y$	等於
!=	$x!=y$	不等於
>	$x>y$	大於
<	$x<y$	小於
>=	$x>=y$	大於等於
<=	$x<=y$	小於等於

- 運算式(expression):
由運算子(operator)
例如==與運算元(operand)例如x或y組成。

練習

寫一個程式讓使用者輸入分數，判斷分數是否大於等於60分，如果是則顯示及格。

輸入範例:

85

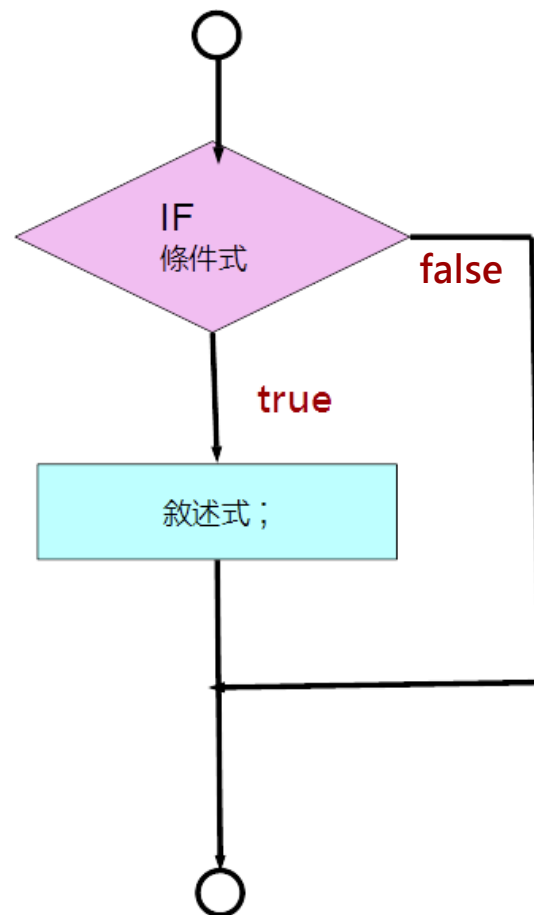
輸出範例:

85分及格



if選擇敘述式流程圖

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



參考程式碼

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int score;
6     cin>>score;
7     if (score>=60)
8     {
9         cout<<score<<"分及格"<<endl;
10    }
11    return 0;
12 }
```

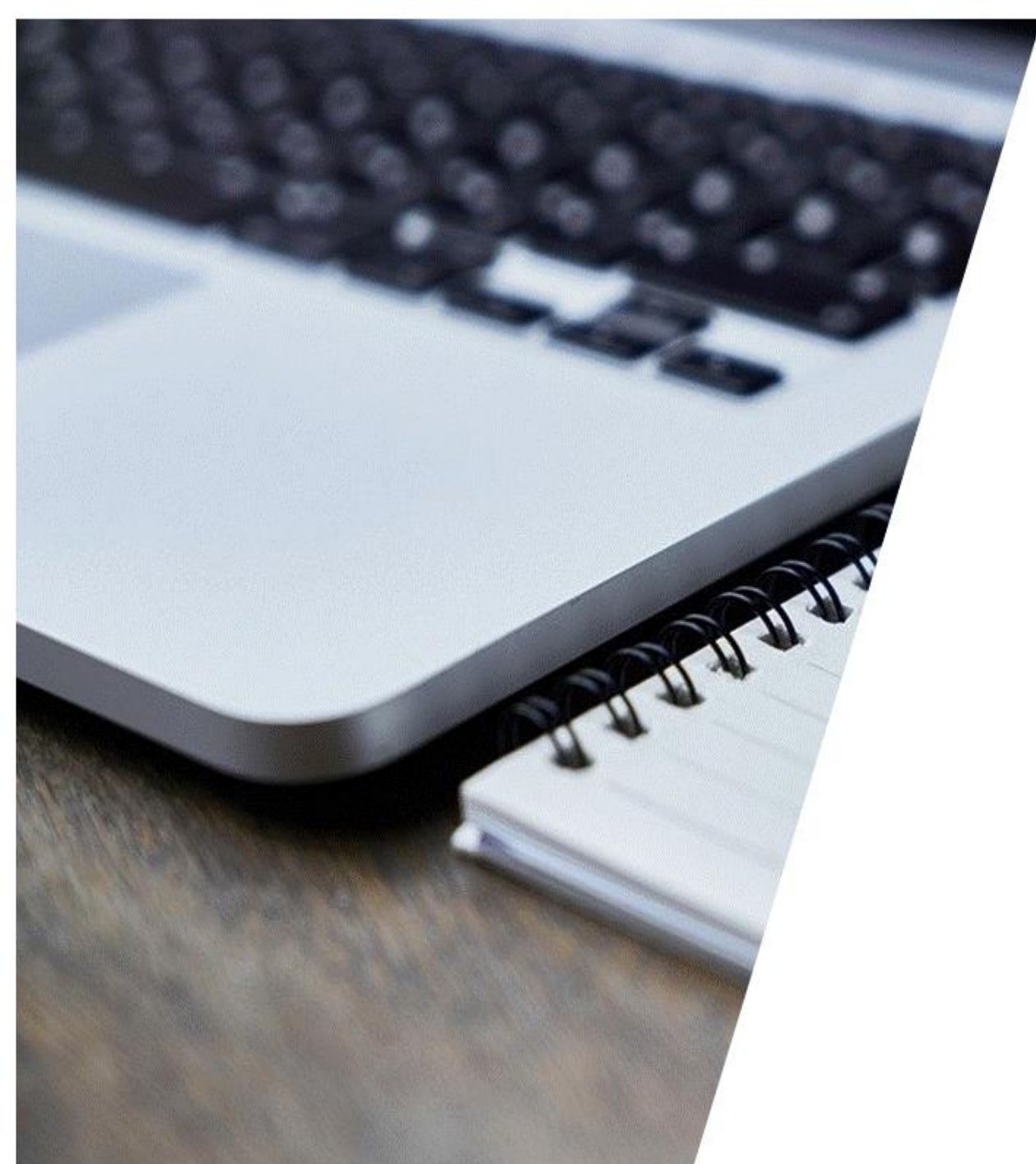
- 如果score>=60，印出分數及格。
- if(score>=60) cout<<score<<"分及格 " 可以寫成一行，如寫法2
- if條件式後面只有一個句子的情形，{}可省略，如寫法3。

```
/*寫法2*/
```

```
if (score>=60) cout<<score<<"分及格"<<endl;
```

```
/*寫法3*/
```

```
if (score>=60)
    cout<<score<<"分及格"<<endl;
```



C ++

延伸的概念

Extended concept



if條件式後面，怎麼沒有分號作為結束？

- 句末要以;(分號)作結束的語法並沒有改變。
- 第6行與第7行是一起的，理論上可以寫成一行：
if(n>0) cout<<n<< “是正數” <<endl;
- 從程式可讀性的觀點來看，還是分成兩行的寫法較佳。

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

int main(){
    int n;
    cin>>n;
    if(n>0)
        cout<<n<<“是正數”<<endl;
    return 0;
}
```

if包含的條件式需要加大括號{ }嗎?

- 如果if後面的條件式只有一行，{}可以被省略。

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

int main(){
    int n;
    cin>>n;
    if(n>0)
        cout<<n<<"是正數"<<endl;
    return 0;
}
```

縮排有利於程式碼外觀與程式碼可讀性

- C 語言是一種格式自由的語言, 編譯器只要求文法正確, 並不要求你用什麼樣子的格式來撰寫程式碼外觀。
- 注意: 雜亂無章的程式碼, 會影響除錯(debug)的效率。

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

int main(){
    int n;
    cin>>n;
    if(n>0)
        cout<<n<<"是正數"<<endl;
    return 0;
}
```

此寫法較好

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

int main(){
    int n;
    cin>>n;
    if(n>0){
        cout<<n<<"是正數"<<endl;
    }
    return 0;
}
```