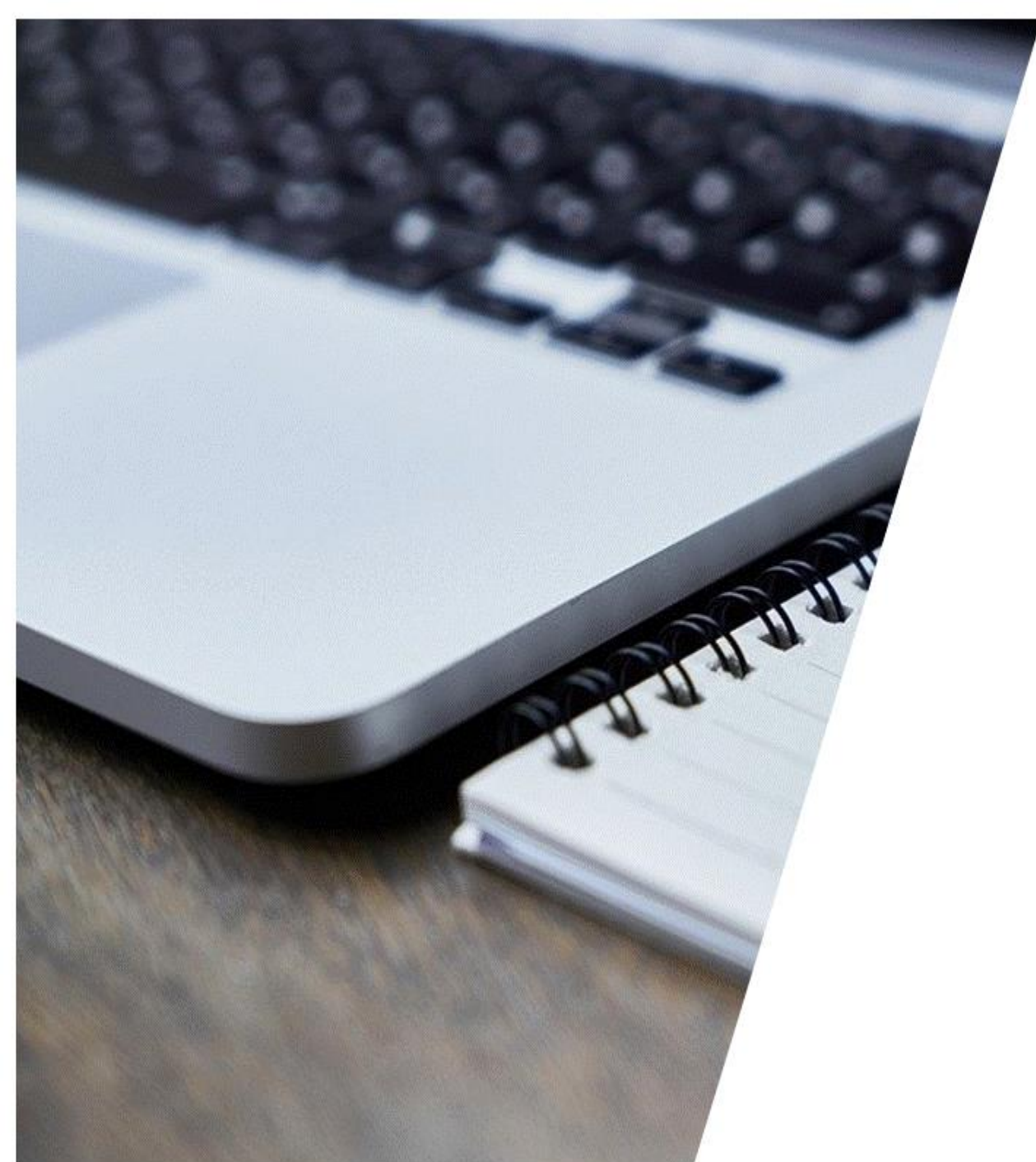




C++

多選一



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

從巢狀選擇選擇演化成多選一的結構，

練習

寫一個程式讓使用者輸入分數，印出分數等級。100
到90之間是等級 A，如果92分，則印出
Your score is 92 and degree is A!
印出其餘類推如下：

89-80 B

79-70 C

69-60 D

59-0 F



```
#include <iostream>
using namespace std;
int main() {
    int x;
    cin >> x;
    if(x >= 90 && x <= 100)
        cout << "Your score is " << x << " and degree is A!\n";
    else
        if(x >= 80)
            cout << "Your score is " << x << " and degree is B!\n";
        else
            if(x >= 70)
                cout << "Your score is " << x << " and degree is C!\n";
            else
                if(x >= 60)
                    cout << "Your score is " << x << " and degree is D!\n";
                else
                    cout << "Your score is " << x << " and degree is F!\n";
    return 0;
}
```

巢狀選擇：分數等第

如果變數大於等於 90，前四個條件都將為真，但只有第一個測試的 `cout` 敘述式會被執行。

在這個 `cout` 執行之後，最外層的 `else` 部分將被跳過。

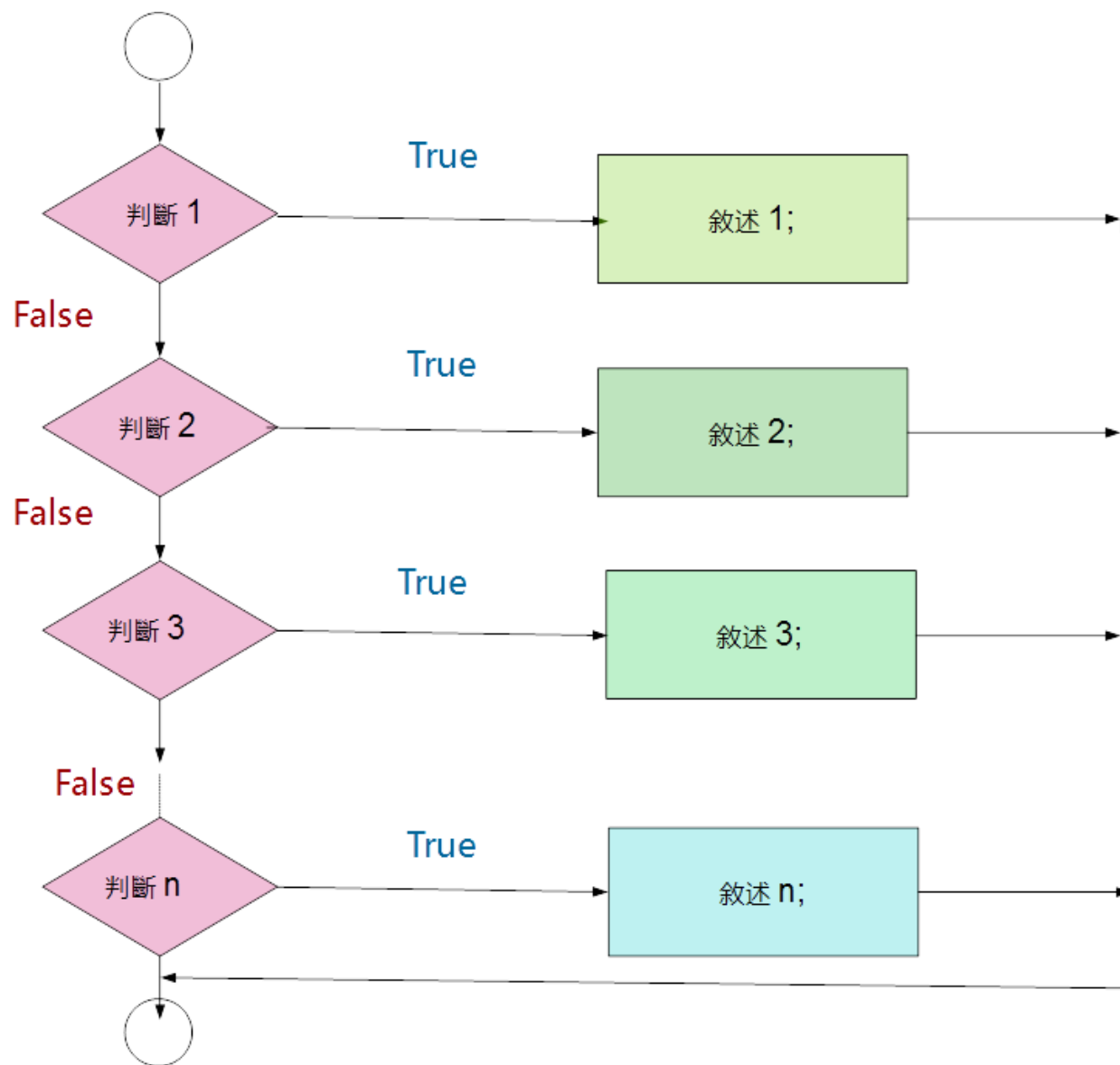
巢狀選擇的演化

```
#include <iostream>
using namespace std;
int main() {
    int x;
    cin >> x;
    if(x >= 90 && x <= 100)
        cout << "Your score is " << x << " and degree is A!\n";
    else if(x >= 80)
        cout << "Your score is " << x << " and degree is B!\n";
    else if(x >= 70)
        cout << "Your score is " << x << " and degree is C!\n";
    else if(x >= 60)
        cout << "Your score is " << x << " and degree is D!\n";
    else
        cout << "Your score is " << x << " and degree is F!\n";
    return 0;
}
```

許多的 C 程式設計師喜歡將上述的if敘述式寫成左邊的if敘述式。

這兩種寫法對C/C++編譯器來說是相同的，而左邊的寫法比較受歡迎的原因是因為避免過身的縮排導致程式向右傾斜，降低程式的可讀性。

多選一示意圖



多選一的語法

if(條件式){

 如果條件成立，執行大括號的句子；

}

else if(條件式){

 在上面的條件不成立的情況下，還有其他的可能，如果符合條件式，則執行大括號的句子；

}

else if(條件式){

 敘述式;

}

else{

 上面的條件不成立的時候，執行這個句子;

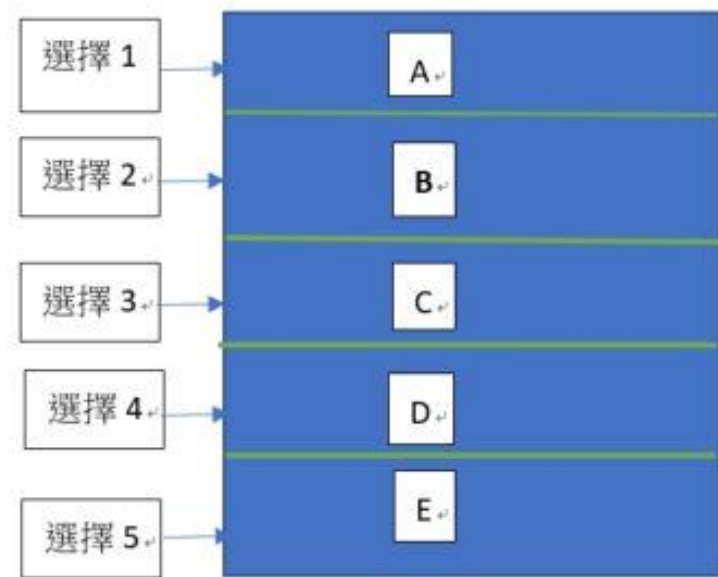
}

程式碼

將可能的選項逐一列出，從第一個條件開始找，找到符合的條件式則執行相對應的敘述式，**不再往下找**。

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

int main(){
    int x;
    cin>>x;
    if(x>=90)
        cout<<"Your score is "<<x<<"and dengree is A!\n";
    else if(x>=80)
        cout<<"Your score is "<<x<<"and dengree is B!\n";
    else if(x>=70)
        cout<<"Your score is "<<x<<"and dengree is C!\n";
    else if(x>=60)
        cout<<"Your score is "<<x<<"and dengree is D!\n";
    else
        cout<<"Your score is "<<x<<"and dengree is E!\n";
    return 0;
}
```



想過這樣寫嗎？

```
#include <iostream>
using namespace std;
int main() {
    int x;
    cin>>x;
    if(x>=90 && x<=100)
        cout<<"Your score is "<<x<<" and degree is A!\n";
    if(x>=80 && x<=89)
        cout<<"Your score is "<<x<<" and degree is B!\n";
    if(x>=70 && x<=79)
        cout<<"Your score is "<<x<<" and degree is C!\n";
    if(x>=60 && x<=69)
        cout<<"Your score is "<<x<<" and degree is D!\n";
    if(x>=0 && x<=59)
        cout<<"Your score is "<<x<<" and degree is F!\n";
    return 0;
}
```

- 如左圖寫法，找到符合條件，執行敘述式之後，**還是會繼續往下比較**。
- 這樣的做法就好比是，你要找一個人的身分證號，x123456789，他藏在A、B、C、D與E的其中一個房間，當你找到 x123456789 的同時，條件比對的工作就應該要停止；如果再繼續比對，結果還是一樣的，浪費的是時間喔。

選擇結構整理

判斷式if

if (條件式)

■ 執行True內容

判斷式if...else

if (條件式)

■ 執行True內容

else

■ 執行False內容

巢狀選擇

if (條件式1)

■ 執行True1內容

else

■ **if** (條件式2)

■ 執行True2內容

■ **else**

■ 執行False內容

多選一

if (條件式1)

■ 執行True1內容

else if 條件式2 :

■ 執行True2內容

else :

■ 執行False內容

淺藍色線段表示縮排，雖然在C/C++中縮排不影響程式的執行，卻影響著程式的可讀性。請養成縮排的好習慣。