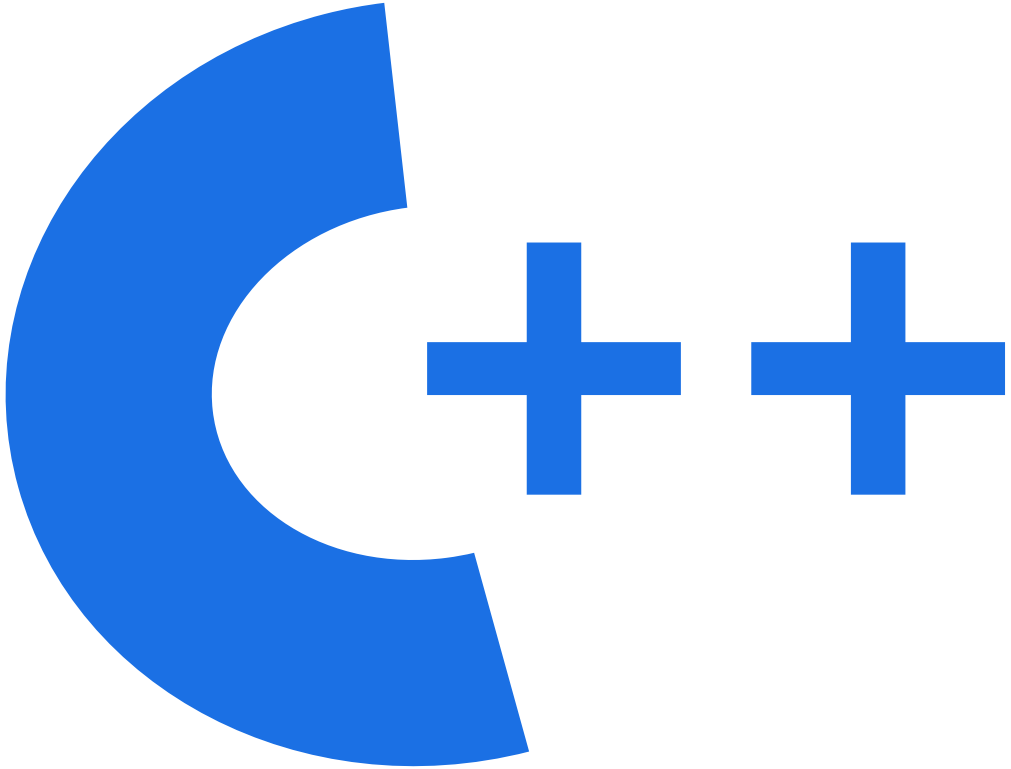


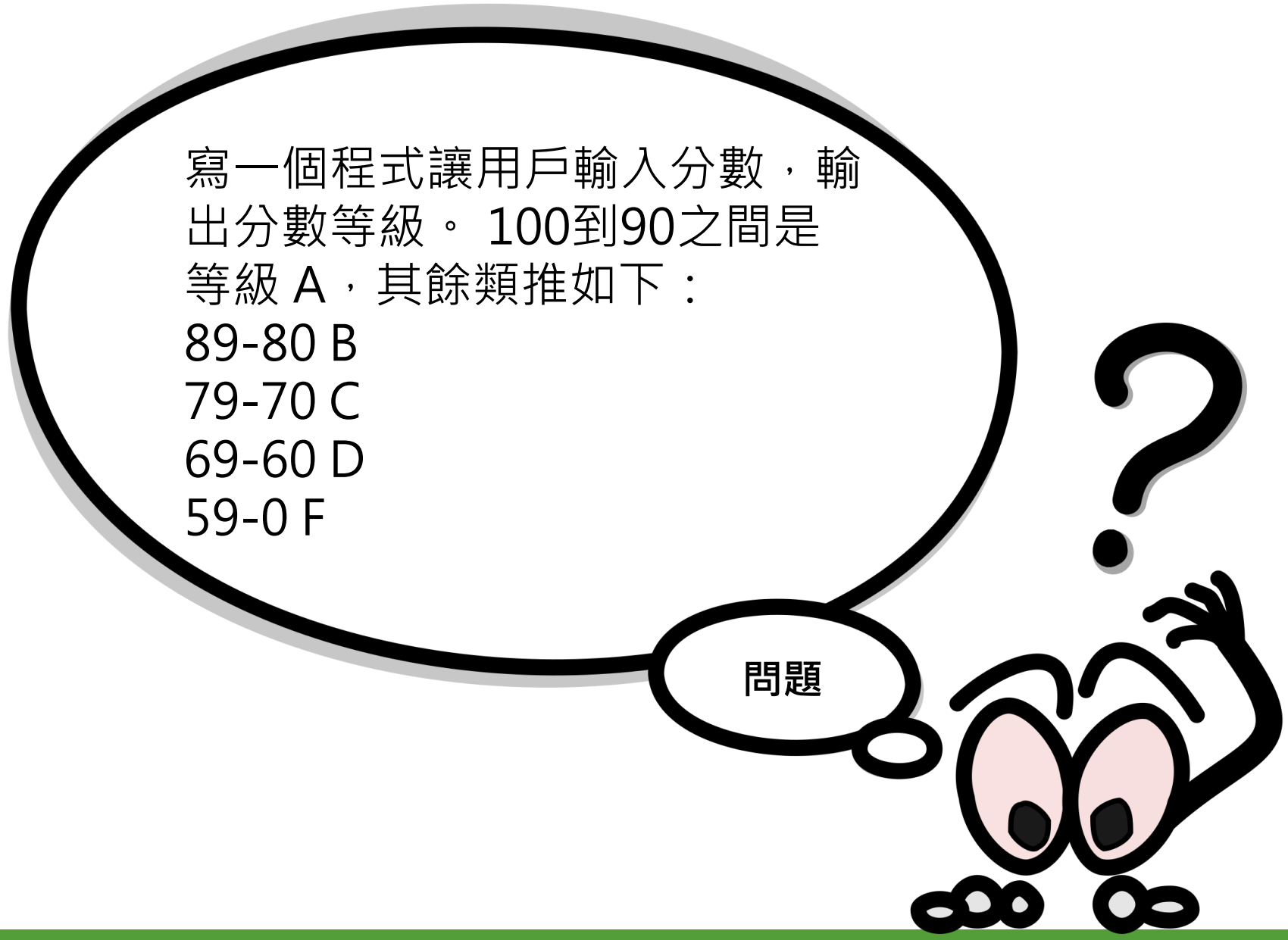
多選一

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，

寫一個程式讓用戶輸入分數，輸出分數等級。 100到90之間是等級 A，其餘類推如下：
89-80 B
79-70 C
69-60 D
59-0 F



巢狀選擇：分數等第

```
if(x>=90)
    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";
```

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printf敘述式會被執行。

在這個printf執行之後，最外層的if...else部分將被跳過。

巢狀選擇的演化

```
if(x>=90)
    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";
```

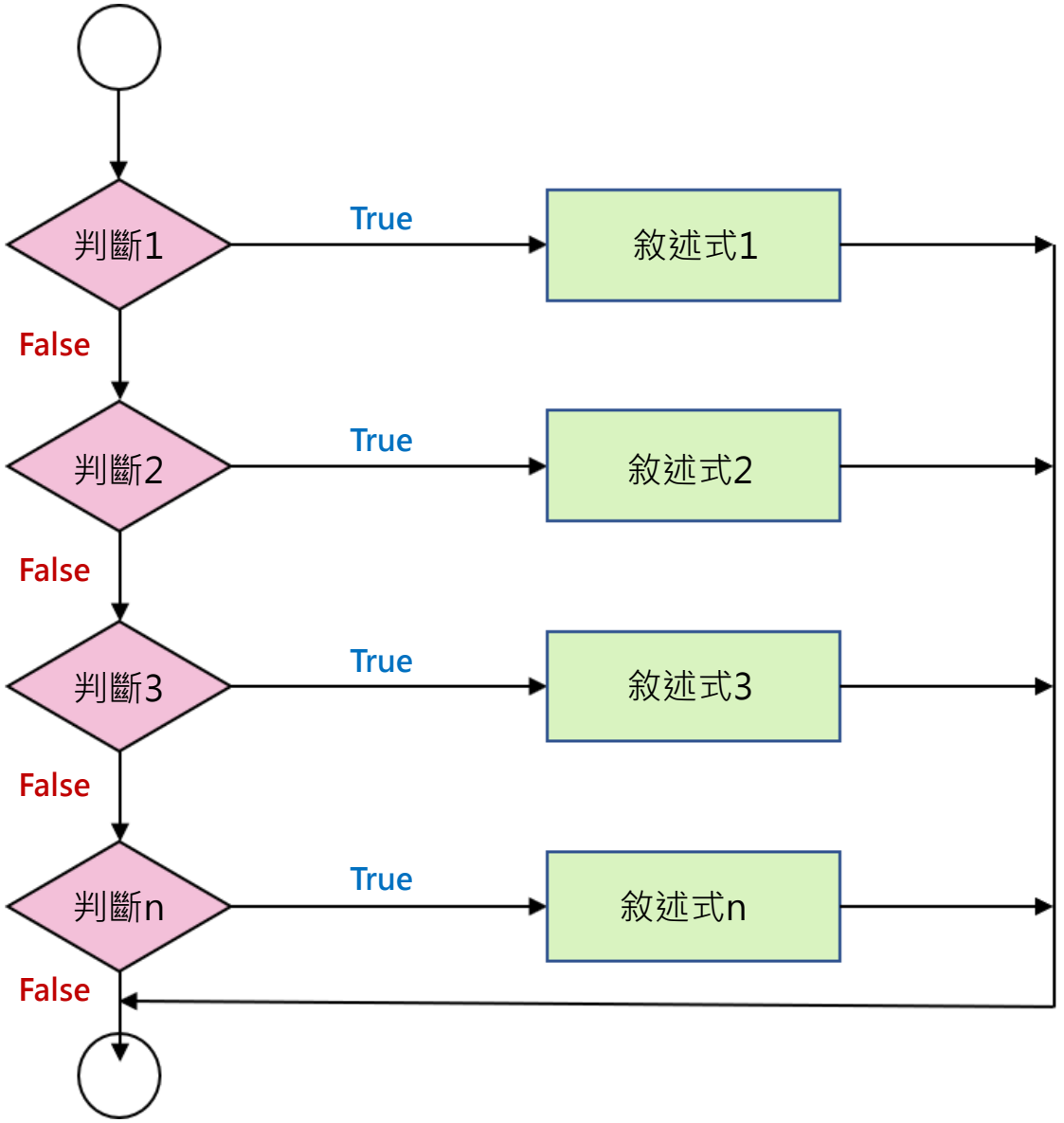
許多的C程式設計師喜歡將上述的if敘述式寫成左邊的if敘述式。

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

多重選擇的虛擬碼

```
if(條件表達式){
        如果條件成立，執行大括號的句子；
}
else if(條件表達式){
        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(條件表達式){
        語句塊;
}
.
.
.
else{
        上面的條件不成立的時候，執行這個句子;
}
```

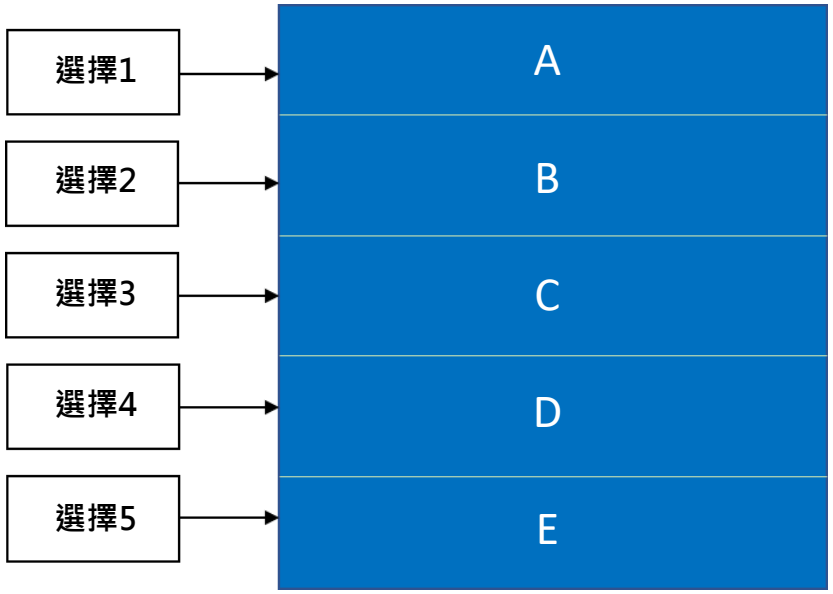
多重選擇示意圖


判斷1
True
敘述式1
False
判斷2
True
敘述式2
False
判斷3
True
敘述式3
False
判斷n
True
敘述式n
False

程式碼

將可能的選項逐一列出，從第一個條件開始找，找到符合的條件表達式則執行對應的語句塊，不再往下找

```
if(x>=90)
    printf("Your score is %d and degree is A!\n",x);
else if(x>=80)
    printf("Your score is %d and degree is B!\n",x);
else if
    (x>=70)printf("Your score is %d and degree is C!\n",x);
else if
    (x>=60)printf("Your score is %d and degree is D!\n",x);
else
    printf("Your score is %d and degree is F!\n",x);
```



想過這樣寫嗎？

```
if(x>=90&&<=100)
    cout<<“A\n”;
if(x>=80&&<=89)
    cout<<“B\n”;
if(x>=70&&<=79)
    cout<<“C\n”;
if(x>=60&&<=69)
    cout<<“D\n”;
if(x>0&&<=59)
    cout<<“F\n”;
```

- 如左圖寫法，找到符合條件，執行語句塊之後，還是會繼續往下比較。
- 這樣的做法就好比是，你要找一個人的身份證字號「x123456789」，他藏在房間A、B、C、D與E的其中一間，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，只是浪費時間。