

流程圖與選擇性敘述


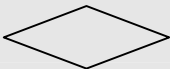

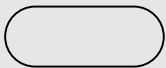
CHAPTER

4

4-1 | 流程圖之表示符號

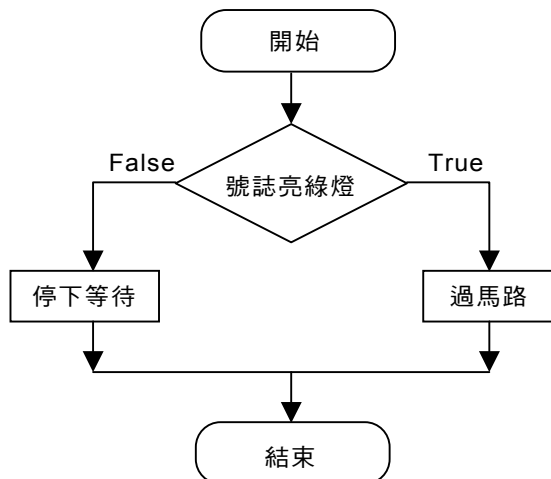
我們現在所使用的流程圖（Flow Chart）是在西元 1940 年由 John von Neumann 所訂定的。在進行較複雜的程式設計時，往往會藉由繪製流程圖，來輔助程式的設計。藉由流程圖的繪製，可以讓程式設計者更清楚地掌握程式的運作流程，並且對於程式的邏輯思考架構，有更深刻的瞭解。

本書從第 4 章開始，就陸續有流程圖出現。這裡先針對繪製流程圖所常使用的流程圖符號，進行列表說明：

流程圖符號	名稱	說明
	程序	代表一個程式區塊，具備執行特定工作的能力。
	決策	依條件式內容來進行條件判斷，以決定程序的下一條路徑。
	流程	用來表示程式的執行方向，具有方向性。
	開始/結束	用來表示演算法的開始或是結束。

我們用下面的例子來說明如何使用流程圖來幫助我們思考與解決問題。例如當我們走到十字路口時，我們需要依據號誌的燈號，做一個判斷動作，以決定是否要過馬路或停下等待。

參考的流程圖如下所示，假如在十字路口，遇到號誌亮綠燈才可以過馬路，否則就停下等待。當號誌亮綠燈時，條件判斷式為真（True），隨後進入過馬路動作，否則，條件判斷式為偽（False），必須進入停下等待動作。



基本上流程圖所表達的是一套我們制定好的規則，並且用圖形來表達，善用它可以讓一些複雜的規則更容易傳達，也可以幫助我們去思考問題的解決方式。使用流程圖的優點在於讓人較容易了解整個作業流程，易於程式的除錯，也有助於程式的修改與維護。本書的程式流程圖為了減少頁面篇幅空間，以下的參考流程圖將會省略開始及結束等流程圖符號。

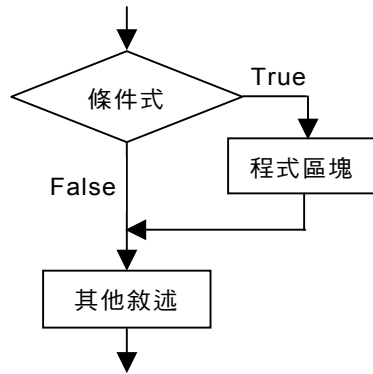
4-2 | 判斷結構之 if 敘述

if 敘述的用途是用來判斷是否進入程式區塊，如果條件式成立，則會進入程式區塊，如果條件式不成立，則會進入 if 敘述之後的其他敘述。其使用的語法如下：

```
if (條件式) {  
    程式區塊;  
}
```



使用流程圖的表示法如如：



if 敘述並不複雜，可以直接解釋為：如果條件式為真，就進入程式區塊執行敘述。參考下面的程式碼：

```

if (a%2==0)
    a=a/2;
    
```

上面的兩行敘述會判斷變數 a 是否為偶數，此處是使用『取餘數』算術運算子，如果餘數等於 (==) 0，則進入 if 敘述將其值除以 2；也就是說如果 a 對 2 取餘數其值等於 0，則表示 a 是 2 的倍數，也代表 a 為偶數。

在上面這個例子中並沒有加上大括號，因為如果只有一行程式敘述，大括號可以省略，如果超過一行，就必須加上大括號。再來看一個類似的例子：

```

if (a>=10 && a<=99) {
    b=a/10;
    c=a%10;
}
    
```

上面這四行敘述（假設變數 a,b,c 為整數型態）會判斷變數 a 是否為二位數，如果 a 為二位數（a 介於 10~99 之間），則 if 敘述會將變數 a 的十位數部分，指定給變數 b，再將變數 a 的個位數部分，指定給變數 c。



程式範例：百貨公司週年慶打折程式

學習重點：熟悉 if 敘述的使用 / 參考檔案：4-2-1.cpp

🔊 程式設計目標

真神氣百貨公司週年慶，公司決定在週年慶期間對消費超過 2000 元的顧客打 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。下圖為輸入『6999』的執行結果。

```

D:\Examples\ch4\4-2-1.exe
請輸入購買金額:6999
實需付4899

-----
Process exited with return value 0
Press any key to continue . . .
    
```

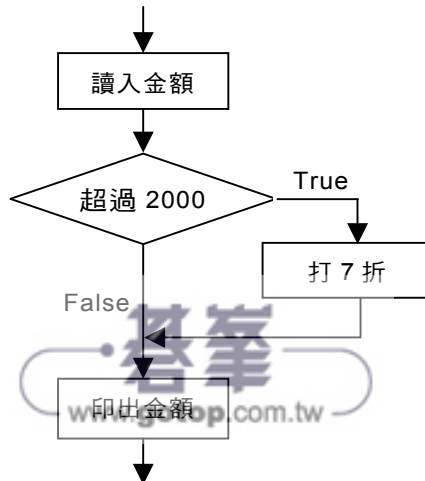
下圖為輸入『1800』的執行結果。

```

D:\Examples\ch4\4-2-1.exe
請輸入購買金額:1800
實需付1800

-----
Process exited with return value 0
Press any key to continue . . .
    
```

參考流程圖：



參考程式碼

```

/*百貨公司週年慶打折程式 */
//4-2-1.cpp
#include <iostream>
using namespace std;
int main()
{
    int money;
    cout<<"請輸入購買金額:";
    cin>>money;                // 輸入買了多少錢
    if(money > 2000)           // 若金額大於兩千
        money = money*7/10;    // 打七折
    cout<<"實需付"<<money<<endl; //印出計算結果

    return 0;
}
    
```

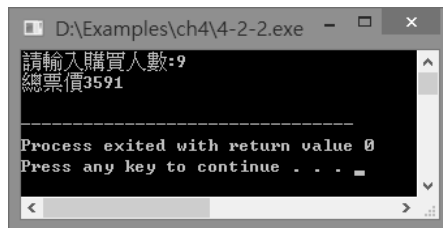


程式範例：遊樂園總票價計算程式

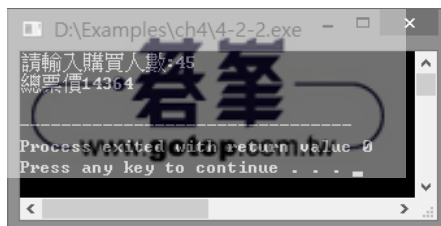
學習重點：熟悉 if 敘述的使用／參考檔案：4-2-2.cpp

程式設計目標

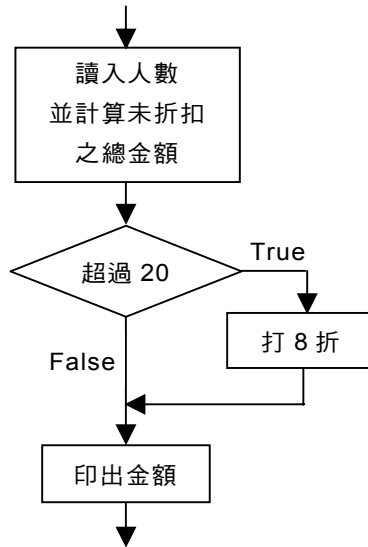
遊樂園為了在暑期促銷，決定只要團體購買人數超過 20 人就可以打 8 折，原價一張票 399 元。請設計一程式輸入為購票人數，程式為計算出總票價。下圖為輸入 9 人的結果：



下圖為輸入 45 人的執行結果。



參考流程圖：



參考程式碼

```

/*遊樂園總票價計算程式*/
//4-2-2.cpp
#include <iostream>
using namespace std;
int main()
{
    int people, money;
    cout<<"請輸入購買人數:";
    cin>>people;
    money=people*399;           // 輸入人數
    if(people > 20)           // 若人數大於二十
        money = money*8/10;   // 打八折
    cout<<"總票價"<<money<<endl; //印出計算結果

    return 0;
}
  
```

4-3 | 判斷結構之 if...else...敘述

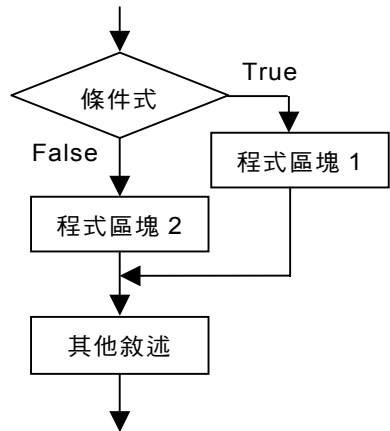
if...else...敘述會根據條件式的真偽，決定要進入的程式區塊為何。如果 if 條件式成立，則會執行 if 部分的『程式區塊 1』；如果不成立則會執行 else 部分的『程式區塊 2』，使用的語法如下：

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

使用流程圖的表示法如右：

if...else...敘述有一個特點，就是在程式區塊 1 和 else 之後的程式區塊 2，其中有一段必會被執行，如果判斷式為真，就執行 if 之後的程式區塊 1，如果為偽，就執行 else 之後的程式區塊 2，參考下面的程式碼：

```
if(a%2)
    cout<<"a is odd";
else
    cout<<"a is even";
```



上面的四行敘述會判斷 a 為偶數或奇數。如為奇數，對 2 取餘數會得到 1，也就是 if 敘述內的條件式為 True，因此會執行 if 下的『cout<<"a is odd";』敘述。如為偶數，對 2 取餘數會得到 0，也就是 if 敘述內的條件式為 False，因此會執行 else 下的『cout<<"a is even";』敘述。

程式範例：雨天帶傘判斷程式

學習重點：熟悉 if...else...敘述的使用 / 參考檔案：4-3-1.cpp

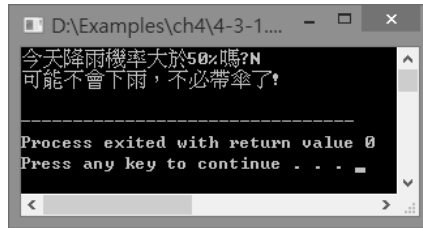


🔊 程式設計目標

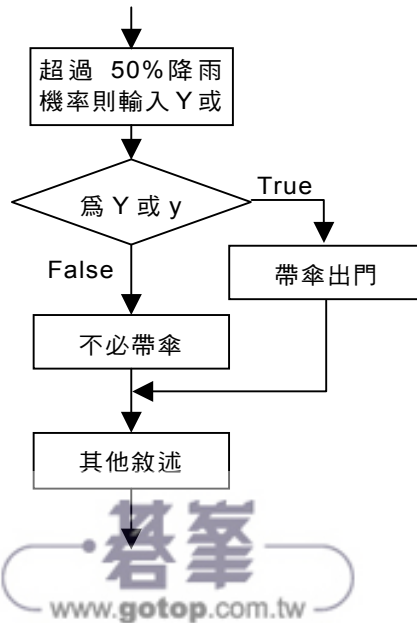
寫出一個程式，請問使用者今天是否有下雨，根據使用者的回答，輸出今天是否應該帶傘。下圖為使用者回答『Y』的執行結果。



下圖為使用者回答『N』的執行結果。



參考流程圖：



🔊 參考程式碼

```
//4-3-1.cpp
#include <iostream>
using namespace std;
int main()
{
    char rain;
    cout<<"今天降雨機率大於 50%嗎?";
    cin>>rain;                // 儲存答案至字元型態變數 rain
    if(rain == 'y' || rain == 'Y') // 判斷輸入是否為 y 或是 Y
        cout<<"你最好要帶傘"<<endl; // 輸入為 Y 或 y 的情況
    else // 當使用者沒有按下 y 或 Y 時，進入 else 敘述中
        cout<<"可能不會下雨，不必帶傘了!"<<endl;

    return 0;
}
```

4-4 | 判斷結構之 if...else if...else...敘述

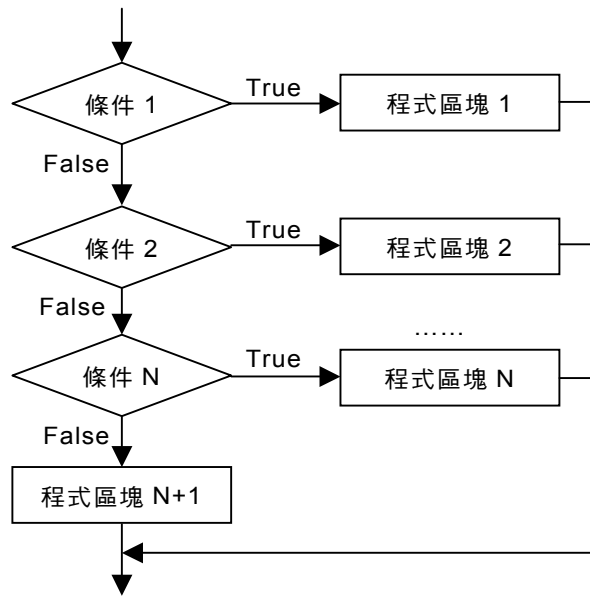
if...else if...else...敘述提供了多種狀況的選擇判斷，根據不同的條件判斷式，一一判斷要進入哪一個程式區塊。

使用的語法如下：

```
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;
}
```



使用流程圖的表示法如下：



參考下面的程式碼，此範例程式會依據條件式來判斷 a 是正數、零或是負數。當 a 值大於 0 時，會印出『 a is positive』；當 a 值等於 0 時，會印出『 a is zero』；當 a 值小於 0 時，則會印出『 a is negative』。

```

if(a>0)
    cout<<"a is positive";
else if (a==0)
    cout<<"a is zero";
else
    cout<<"a is negative";
  
```



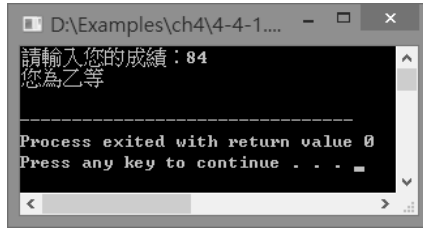
程式範例：成績區間判斷程式

學習重點：熟悉 if...else if...else...敘述的使用／參考檔案：4-4-1.cpp

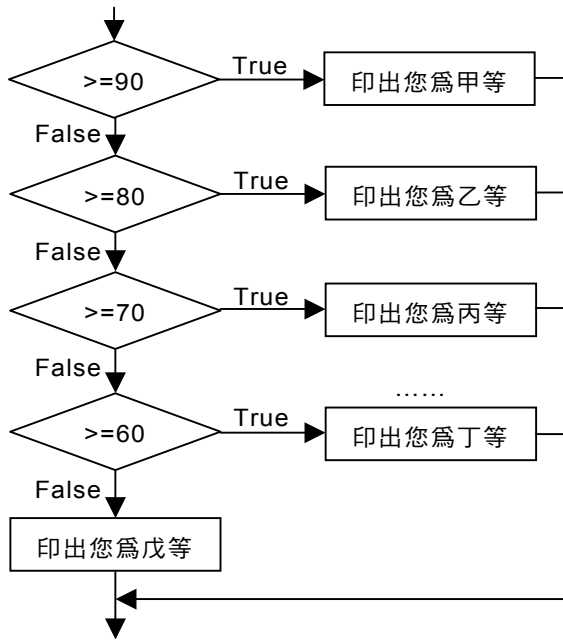
🔊 程式設計目標

依據使用者輸入的成績，判斷其成績等第為何？90 分以上為甲等，80 分到未滿 90 分之間為乙等，70 分到未滿 80 分之間為丙等，60 分到未滿 70

分之間為丁等，未滿 60 分為戊等。下圖為輸入『84』的執行結果，程式會告知使用者『您為乙等』。



參考流程圖：



參考程式碼

```

/* 成績區間判斷程式 */
//4-4-1.cpp
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout<<"請輸入您的成績："
    cin>>score;
    //將成績存入整數變數 score 中

```



```
if(score >= 90)                //是否為 90 分以上
    cout<<"您為甲等"<<endl;
else if(score >= 80)           //小於 90 但 80 以上
    cout<<"您為乙等"<<endl;
else if(score >= 70)           //小於 80 但 70 以上
    cout<<"您為丙等"<<endl;
else if(score >= 60)           //小於 70 但 60 以上
    cout<<"您為丁等"<<endl;
else                            //小於 60 分
    cout<<"您為戊等"<<endl;

return 0;
}
```

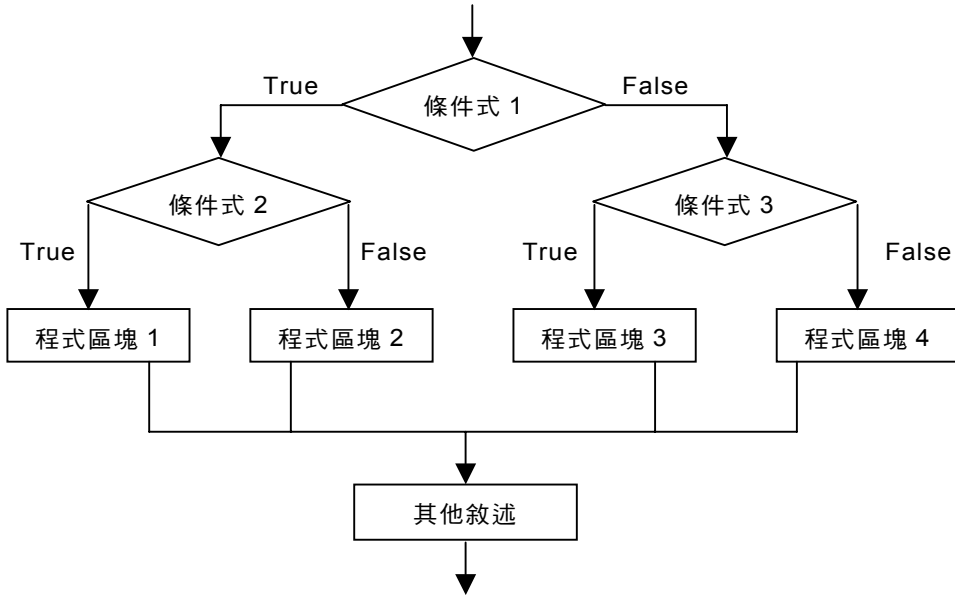
4-5 | 判斷結構之巢狀 if

當 if 敘述的程式區塊中包含了其他的 if 敘述時，我們稱此種 if 敘述為巢狀 if 敘述，我們舉一個巢狀 if 敘述的語法架構如下：

```
if(條件式 1) {
    if(條件式 2) {
        程式區塊 1;
    }
    else {
        程式區塊 2;
    }
else {
    if(條件式 3) {
        程式區塊 3;
    }
    else {
        程式區塊 4;
    }
}
```



使用流程圖的表示法如下：



當我們在撰寫巢狀 if 的時候，盡量將相對應的 if...else...敘述對齊，以利未來程式之維護或修改。

程式範例：閏年判斷程式

學習重點：熟悉巢狀 if 敘述的使用 / 參考檔案：4-5-1.cpp

🔊 程式設計目標

請寫出一程式，輸入西元年份，判斷該年是否為閏年(閏年判斷方式為：四年一閏，逢百年不閏，四百年又閏)。如果輸入『2100』，程式會輸出『2100年不為閏年』。如果輸入『2000』，程式會輸出『2000年為閏年』。

如圖為輸入『480』的執行結果。



🔊 參考程式碼

```
//4-5-1.cpp
#include <iostream>
using namespace std;
int main()
{
    int year;
    cin>>year;
    if(!(year%4)) {                //可被 4 整除
        if(!(year%100) && year%400) //被 4 和 100 都整除，不被 400 整除
            cout<<year<<" 年不為閏年"<<endl;
        else
            cout<<year<<" 年為閏年"<<endl; //被 4 和 400 都整除，不被 100 整除
    }
    else
        cout<<year<<" 年不為閏年"<<endl;    //無法被 4 整除

    return 0;
}
```



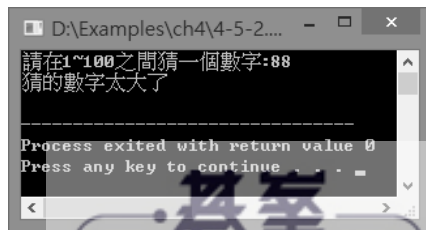
程式範例：猜數字遊戲

學習重點：熟悉巢狀 if 敘述的使用／參考檔案：4-5-2.cpp

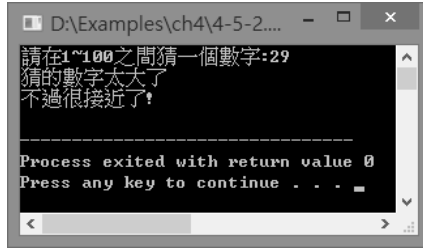
🔊 程式設計目標

請寫出一程式，使用者試著猜出 1~100 之間的一個數字，當猜的數字太大時，告訴使用者他猜得太大了，反之亦然。而當他猜的數字很接近，差距在 5 以內時，則告訴他已經接近答案了。假設本題的解答為 27。

如圖為猜測其數字為『88』，程式會回應『猜的數字太大了』。



如圖為猜測其數字為『29』，程式會回應『猜的數字太大了，不過很接近了!』。



如圖為猜測其數字為『27』，程式會回應『恭喜你猜對了！』。



參考程式碼

```
//4-5-2.cpp
#include <iostream>
using namespace std;
int main()
{
    int answer=27; //設定數字解答
    int guess=-1;
    cout<<"請在 1~100 之間猜一個數字:";
    cin>>guess;
    if( guess==answer ) cout<<"恭喜你猜對了!"<<endl; //猜中了
    else if ( guess<answer ){ //猜的數字小於解答
        cout<<"猜的數字太小了"<<endl;
        if(answer-guess<=5) cout<<"不過很接近了!"<<endl; //和解答很接近
    }
    else {
        cout<<"猜的數字太大了"<<endl; //猜的數字大於解答
        if(guess-answer<=5) cout<<"不過很接近了!"<<endl; //和解答很接近
    }

    return 0;
}
```



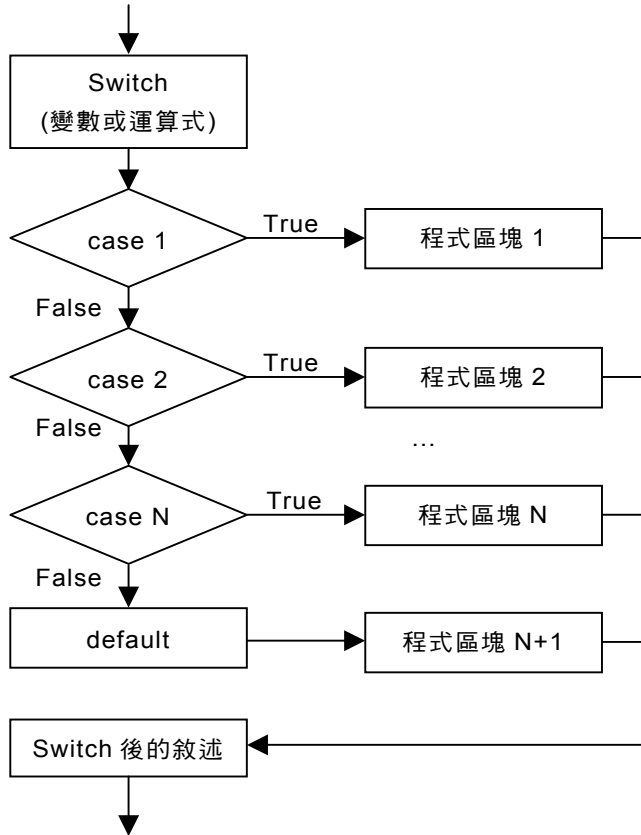
4-6 | 判斷結構之 switch 敘述

想要提供兩個選擇時，我們可以用 `if` 來做判斷，選擇要執行的程式區塊。當有多重選擇要進行判斷，要多選一時，我們可以用 `if...else if...else...` 敘述來做判斷。另外，C++ 語言為了將多選一的情況簡化，還提供了 `switch` 敘述，`switch` 會根據某一個字元或是整數變數，來判斷要進入哪一個程式區塊，適用的語法如下：

```
switch(變數或運算式) {  
    case 值 1:  
        程式區塊 1;  
        break;  
    case 值 2:  
        程式區塊 2;  
        break;  
    ...  
    case 值 N:  
        程式區塊 N;  
        break;  
    default:  
        程式區塊 N+1;  
}
```



也可以用下方的流程圖表示：



switch 敘述會根據放在括號內的變數或是運算式的結果，比對下方 case 內的資料，如果有 case 指定的值符合，就執行該程式區段，直到遇到 break 敘述，再跳出 switch 的程式區塊；如果沒有值符合，會進入 default 之後的程式區塊。有時也可以不寫 default 區塊，直接跳出 switch 區塊。

我們先來看一個模擬計算機程式，我們先讓使用者輸入想要執行的運算（以一個字元表示），如果輸入『+』符號，程式會計算將變數 a 的值加上變數 b 的值，並將結果印出；如果輸入『-』符號，程式會計算將變數 a 減去 b 的值，並將結果印出；如果輸入的不是『+』也不是『-』符號，程式印出『I don't know how to do that.』字串，參考程式碼如下。



```

char ch;
cin>>ch;
switch(ch) {
    case '+':
        cout<<"a + b ="<<a+b;
        break;
    case '-':
        cout<<"a - b ="<<a-b;
        break;
    default:
        cout<<"I don't know how to do that.";
}

```

小提示 使用 switch 敘述，最常犯的錯誤是在一個 case 結束時，忘了加上 break 敘述，這時候程式會繼續往下一個 case 執行，直到遇見 break 敘述為止。造成我們只想執行某一個 case 內的程式區塊，卻不小心多執行了其他區塊。初學 switch 敘述時，最好是養成一個 case 對應一個 break 的好習慣。

延伸上一個程式，我們設計一個程式，此程式當使用者輸入『p』或是『+』，都可以將變數 a 和變數 b 相加。當 ch 為 + 時，會進入 case '+' 區塊之中，因為該區塊沒有 break 敘述，就繼續往下執行，接著會進入 case 'p' 區塊之中，印出 a+b 的結果之後，才遇到 break 敘述，跳出 switch 程式區塊，結束 switch 敘述的執行。

```

switch(ch) {
    case '+':
    case 'p': //plus
        cout<<"a + b ="<<a+b;
        break;
    case '-':
    case 'm': //minus
        cout<<"a - b ="<<a-b;
        break;
}

```



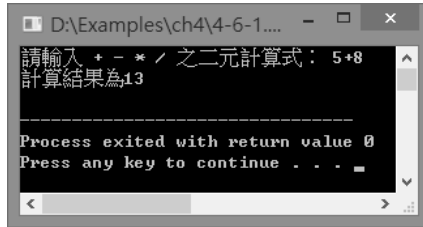


程式範例：運用 switch 敘述設計四則計算機程式

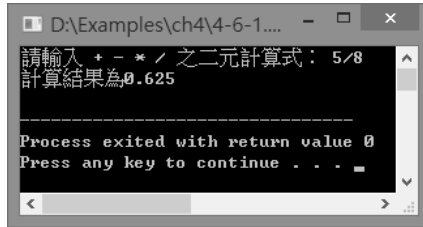
學習重點：熟悉 switch 敘述的使用 / 參考檔案：4-6-1.cpp

程式設計目標

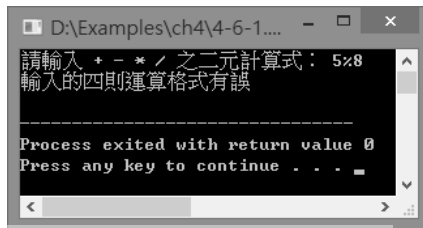
用 switch 敘述撰寫一個做四則計算的程式，可以處理兩個運算元和一個加、減、乘、除運算子的計算。例如：輸入『5+8』，程式會回應『13』，執行結果如下圖所示。



輸入『5/8』，程式會回應『0.625000』，執行結果如下圖所示。



輸入『5%8』，程式會回應『輸入的四則運算格式有誤』，執行結果如下圖所示。



🔊 參考程式碼

```

/* 用 Switch 敘述做四則計算程式 */
//4-6-1.cpp
#include <iostream>
using namespace std;
int main()
{
    int a,b;           //用來儲存使用者輸入的數值
    char oper;        //用來存使用者輸入的計算子+ - * /
    cout<<"請輸入 + - * / 之二元計算式： ";
    cin>>a>>oper>>b;
    switch(oper)      //利用 switch 判斷計算子為+-*/中哪一項
    {
        case '+':     //若計算子為+
            cout<<"計算結果為"<<a+b<<endl;
            break;
        case '-':     //若計算子為-
            cout<<"計算結果為"<<a-b<<endl;
            break;
        case '*':     //若計算子為*
            cout<<"計算結果為"<<a*b<<endl;
            break;
        case '/':     //若計算子為/
            cout<<"計算結果為"<<(float)a/b<<endl; //此處強制轉換形態為浮點數
            break;
        default:      //若計算子不為+-*/中其中一項
            cout<<"輸入的四則運算格式有誤"<<endl;
    }

    return 0;
}

```

4-7 | 程式練習

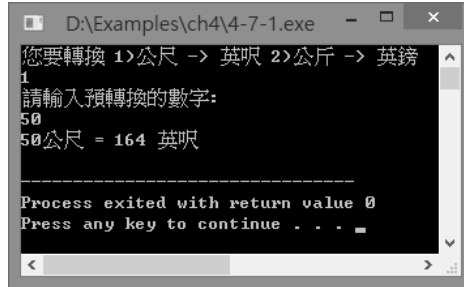


程式練習 1：單位轉換程式

學習重點：熟悉 if...else if...else... 敘述的使用 / 參考檔案：4-7-1.cpp

🔊 程式設計目標

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『公尺->英呎』或『公斤->英鎊』的程式。下圖為輸入『1』後，選擇公尺轉英呎運算，再輸入『50』的執行結果。



🔊 參考程式碼

```
01: /*單位轉換程式 */
02: //4-7-1.cpp
03: #include <iostream>
04: using namespace std;
05: int main(void)
06: {
07:     char ch;           //宣告儲存使用者選擇的變數
08:     double input;     //宣告儲存使用者輸入的數字
09:     cout<<"您要轉換 1)公尺 -> 英呎 2)公斤 -> 英鎊"<<endl;
10:     cin>>ch;          /* 輸入選擇 */
11:     cout<<"請輸入預轉換的數字:"<<endl;
12:     cin>>input;       /* 輸入欲轉換的數 */
13:     if(ch == '1')     //轉換公尺
14:         cout<<input<<"公尺 = "<<input*3.28<<" 英呎"<<endl;
15:     else if(ch == '2') //轉換公斤
16:         cout<<input<<"公斤 = "<<input*2.2<<" 英鎊"<<endl;
17:     else
18:         cout<<"沒有這個選項"<<endl;
19:
20:     return 0;
21: }
```



程式練習 2：季節判斷程式

學習重點：熟悉 if...else if...else...敘述的使用／參考檔案：4-7-2.cpp

🔊 程式設計目標

我們都知道一年有四季，分別是春（1～3月）、夏（4～6月）、秋（7～9月）、冬（10～12月），請寫一個程式判斷輸入的月數是什麼季節。如果輸入『2』，程式會回應『2月是春天』，執行結果如下圖所示。

🔊 參考程式碼

```

01: /* 季節判斷程式 */
02: //4-7-2.cpp
03: #include <iostream>
04: using namespace std;
05: int main(void)
06: {
07:     int month;
08:     cout<<"請輸入月份：";
09:     cin>>month;
10:     if(month >= 1 && month <=3)           //判斷是否為 1 月到 3 月
11:         cout<<month<<" 月是春天"<<endl;
12:     else if(month <=6)                   //判斷是否為 4 月到 6 月
13:         cout<<month<<" 月是夏天"<<endl;
14:     else if(month <=9)                   //判斷是否為 7 月到 9 月
15:         cout<<month<<" 月是秋天"<<endl;
16:     else if(month <=12)                  //判斷是否為 10 月到 12 月
17:         cout<<month<<" 月是冬天"<<endl;
18:     else
19:         cout<<"不合法的月份"<<endl;
20:
21:     return 0;
22: }

```





程式練習 3：購物計費程式

學習重點：熟悉 if...else if...else...敘述的使用 / 參考檔案：4-7-3.cpp

🔊 程式設計目標

獵人小傑拿了一些錢，去便利商店買得意的一天葵花油、愛之味山藥麵筋、熊寶貝衣物香氛袋等個數不等的商品，請問他剩下多少錢？

參考價目表如下：

商品名稱	售價（元）
得意的一天葵花油	199
愛之味山藥麵筋	23
熊寶貝衣物香氛袋	85
商品名稱	售價（元）

輸出入要求：輸入小傑帶了 M 元，以及三項物品所要買的數量 a_1, a_2, a_3 ，若錢足夠則買齊所有的項目，輸出"剩下 X 元"，若錢不夠的話，則輸出"還差 X 元"。

下圖為輸入『500』和『1 2 2』的執行結果。

```

D:\Examples\ch4\4-7-3...
500
1 2 2
剩下 85元

-----
Process exited with return value 0
Press any key to continue . . .
    
```



下圖為輸入『300』和『1 2 3』的執行結果。

```

D:\Examples\ch4\4-7-3... - □ ×
300
1 2 3
還差 200元

-----
Process exited with return value 0
Press any key to continue . . .

```

參考程式碼

```

01: //購物計費程式
02: //4-7-3.cpp
03: #include <iostream>
04: using namespace std;
05: int main(void)
06: {
07:     int m,a1,a2,a3,total; //宣告 5 個整數變數
08:     cin>>m>>a1>>a2>>a3; //輸入所攜錢數及商品個數
09:     total=a1*199+a2*23+a3*85; //計算購買的商品總價
10:     if(total<=m) //判斷是否有足夠的錢數
11:         cout<<"剩下 "<<m-total<<"元"<<endl; //所需錢<=擁有錢
12:     else
13:         cout<<"還差 "<<total-m<<"元"<<endl; //所需錢 > 擁有錢
14:
15:     return 0;
16: }

```



程式練習 4：多段式百貨打折程式

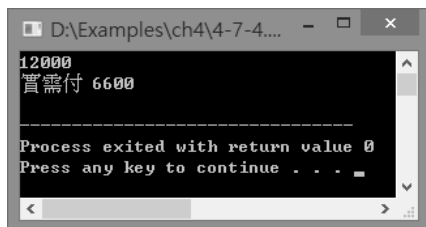
學習重點：熟悉 if...else if...else...敘述的使用／參考檔案：4-7-4.cpp

程式設計目標

真神氣百貨公司週年慶的打折策略，吸引許多客人上門，因此公司決定再回饋客戶，當客戶消費到 2000 元時打 7 折，消費到 5000 元時打 6 折，消費至 10000 元時打 55 折，請幫該公司寫出一個新的收銀台程式，輸入顧客購買總金額後，計算顧客實際需付的錢。



下圖為輸入『12000』的執行結果。



🔊 參考程式碼

```

01: /*多段式百貨公司程式 */
02: //4-7-4.cpp
03: #include <iostream>
04: using namespace std;
05: int main(void)
06: {
07:     int money;
08:     cin>>money;    /* 輸入買了多少錢*/
09:     if(money > 10000)
10:         money = money*55/100;
11:     else if(money > 5000)
12:         money = money*6/10;
13:     else if(money > 2000)
14:         money = money*7/10;
15:     cout<<"實需付 " <<money<<endl;
16:
17:     return 0;
18: }
    
```



程式練習 5：輸入字元的判斷程式

學習重點：熟悉字元變數與 ASCII 碼的使用／參考檔案：4-7-5.cpp

🔊 程式設計目標

請寫一個程式，接受一個字元的輸入，判斷該字元為大寫英文字母、小寫英文字母、阿拉伯數字。



下圖為輸入『6』的執行結果。

```

D:\Examples\ch4\4-7-5...
6
輸入字元為阿拉伯數字
-----
Process exited with return value 0
Press any key to continue . . .
  
```

下圖為輸入『A』的執行結果。

```

D:\Examples\ch4\4-7-5...
A
輸入字元為大寫字母
-----
Process exited with return value 0
Press any key to continue . . .
  
```

下圖為輸入『b』的執行結果。

```

D:\Examples\ch4\4-7-5...
b
輸入字元為小寫字母
-----
Process exited with return value 0
Press any key to continue . . .
  
```

參考程式碼

```

01: /*字元判斷程式 */
02: //4-7-5.cpp
03: #include <iostream>
04: using namespace std;
05: int main(void)
06: {
07:     char ch;
08:     cin>>ch;
09:     if(ch >= 'a' && ch <= 'z')
10:         cout<<"輸入字元為小寫字母\n";
11:     if(ch >= 'A' && ch <= 'Z')
12:         cout<<"輸入字元為大寫字母\n";
13:     if(ch >= '0' && ch <= '9')
  
```

```

14:         cout<<"輸入字元為阿拉伯數字\n";
15:
16:     return 0;
17: }
    
```

🔊 程式碼解說

- ◆ 第 8 行：使用 cin 讀入使用者數入的字母，存入 ch 變數中。
- ◆ 第 9~14 行：char 字元型態變數可轉換為 ASCII 碼比較大小，在 ASCII 碼中，大小寫字母為連續的。ASCII 碼表請參考本書附錄。
- ◆ 第 9、10 行：處理小寫字母的情況。
- ◆ 第 11、12 行：處理大寫字母的情況。
- ◆ 第 13、14 行：處理阿拉伯數字的情況。



程式練習 6：解一元二次方程式之 C++ 程式

學習重點：算術運算子與判斷敘述的使用／參考檔案：4-7-6.cpp

🔊 程式設計目標

本範例希望設計一個可以解數學上的一元二次方程式的程式。讓使用者直接輸入一元二次方程式的係數，然後會得到其解的狀況。我們知道一元二次方程式的解之判斷法如下：

假設一元二次方程式為： $ax^2+bx+c=0$

其中， a, b, c 是係數，而一元二次方程式的解為 x 。其解為實根或虛根的判斷方式，依據 b^2-4ac 的值而定，其情況可分為下列三種：

- $b^2-4ac=0$ ：代表此一元二次方程式有兩等根，其值為 $\frac{-b}{2a}$ 。
- $b^2-4ac>0$ ：代表此一元二次方程式有兩相異實根，其值為 $\frac{-b+\sqrt{b^2-4ac}}{2a}$ 及 $\frac{-b-\sqrt{b^2-4ac}}{2a}$ 。
- $b^2-4ac<0$ ：代表此一元二次方程式有兩虛根。



例如：以一元二次方程式 $x^2-5x+6=0$ 為例，其解為有兩個相異實根：3 和 2。下圖為執行結果畫面。

參考程式碼

```

01: // 解一元二次方程式：ax^2 + bx + c = 0 輸入 a, b, c 求解 x
02: //4-7-6.cpp
03: #include <iostream>
04: #include <math.h>
05: using namespace std;
06: int main ()
07: {
08:     double a, b, c;
09:     cout << "ax^2 + bx + c = 0\n 請依序輸入 a, b, c\n";
10:     cin >> a >> b >> c;
11:     cout << "答案是";
12:     if(pow(b, 2) - 4 * a * c > 0)
13:         cout << ((-1) * b + sqrt(pow(b, 2) - 4 * a * c)) / 2 / a <<
            " " <<
14:         (-b - sqrt(pow(b, 2) - 4 * a * c)) / 2 / a << endl;
15:     else if(pow(b, 2) - 4 * a * c == 0)
16:         cout << (-1) * b / 2 / a << endl;
17:     else
18:         cout << "無解\n";
19:     return 0;
20: }

```

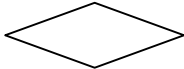



程式碼解說

- ◆ 第 1 行：程式碼的第 1 行是程式的註解。
- ◆ 第 3 行：含括（#include）進來 iostream 標頭檔。
- ◆ 第 4 行：含括（#include）進來數學函式 math.h 標頭檔。
- ◆ 第 5 行：使用 using 指令來宣告要使用 std 命名空間。

- ◆ 第 6~20 行：為 main()函式。
- ◆ 第 12 行：依據 b^2-4ac 的值來判斷解的狀況。
- ◆ 第 13 行：使用 sqrt()函式來求平方根值。
- ◆ 第 17~18 行： b^2-4ac 的值小於 0，為無解的狀況。
- ◆ 第 19 行：程式順利結束，回傳 0 給 main()函式。

4-8 | 習題

選擇題

- () 1. 使用 switch 多重選擇結構，我們通常會在一個 case 結束時，加上何種指令？
- (A) more (B) break
(C) then (D) continue
- () 2. 請問在流程圖中，「決策」所使用的符號為下列何者？
- (A)  (B) 
(C)  (D) 
- () 3. 請問下列哪一種並非 C++的判斷流程語法？
- (A) switch...case (B) select...case
(C) if...else (D) if...else if...else
- () 4. 如果 a 的值為 2，則執行下列程式後會輸出何值？

```
if(a==3)
a=3;
a=4;
```

- (A) 2 (B) 3
(C) 4 (D) 5



() 5. 如果 a 的值為 2，則執行下列程式後 a 為多少？

```
if (a==3)
    a=3;
else
    a=4;
```

- (A) 2 (B) 3
(C) 4 (D) 5

() 6. 下列何者不是流程圖的優點？

- (A) 易於程式的除錯 (B) 讓人較容易了解整個作業流程
(C) 有助於程式的修改與維護 (D) 流程圖符號沒有標準較彈性

問答題

1. 請說明 if 敘述的語法與意涵。
2. 請繪出 C++ 語言 switch 敘述的流程圖。