

ACPS C++ 教學講義 04

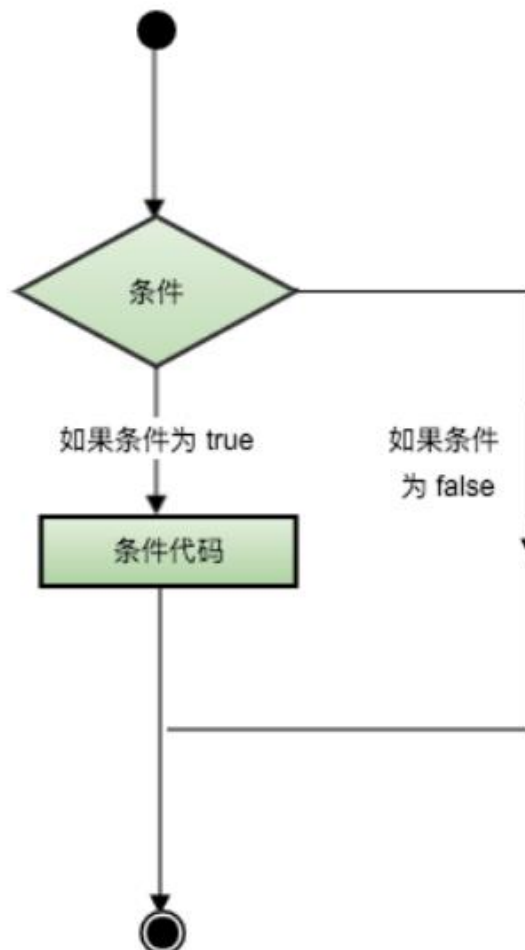
C++ 判斷

判斷結構要求程式設計師指定一個或多個要評估或測試的條件，

以及條件為真時要執行的語句（**必需的**）

和條件為假時要執行的語句（**可選的**）。

下面是大多數程式語言中典型的判斷結構的一般形式：



判斷語句

C++ 程式語言提供了以下類型的判斷語句。

語句	描述
if 語句	一個 if 語句 由一個布林代數後跟一個或多個語句組成。
if...else 語句	一個 if 語句 後可跟一個可選的 else 語句，else 語句在布林代數為假時執行。
巢狀 if 語句	您可以在一個 if 或 else if 語句內使用另一個 if 或 else if 語句。
switch 語句	一個 switch 語句允許測試一個變數等於多個值時的情況。
巢狀 switch 語句	您可以在一個 switch 語句內使用另一個 switch 語句。

if 語句

一個 if 語句 由一個布林代數後跟一個或多個語句組成。

語法

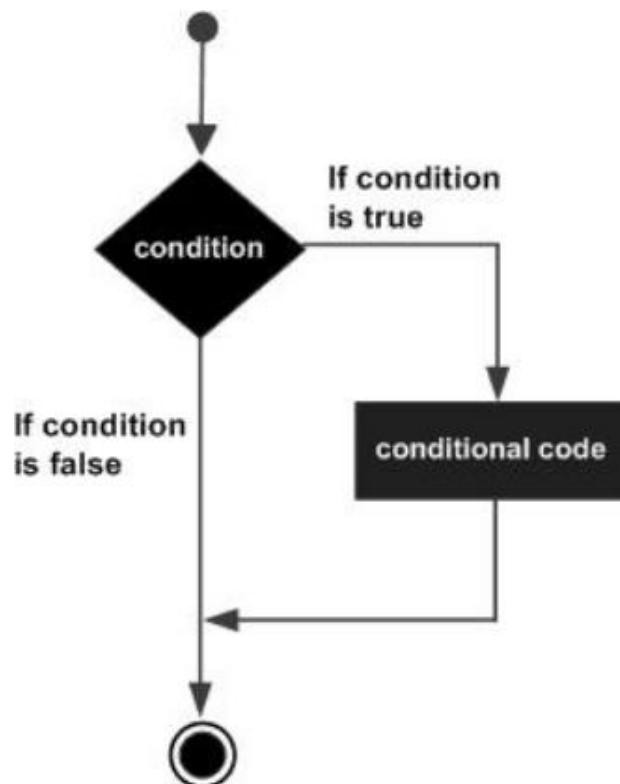
```
if(boolean_expression)
{
    // 如果布林代數為真將執行的語句
}
```

如果布林代數為 true，則 if 語句內的程式碼塊將被執行。

如果布林代數為 false，則跳過 if 語句。

C 語言把任何非零和非空的值假定為 true，把零或 null 假定為 false。

流程圖



實例 4-01

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <iostream>
4  using namespace std;
5
6  int main ()
7  {
8      int a = 10;
9
10     // 使用 if 語句檢查布林條件
11     if( a < 20 ) {
12         // 如果條件為真，則輸出下面的語句
13         cout << "a 小於 20" << endl;
14     }
15     cout << "a 的值是 " << a << endl;
16
17     return 0;
18 }
```

當上面的程式碼被編譯和執行時，它會產生下列結果：

```
a 小於 20  
a 的值是 10
```

if...else 語句

一個 if 語句 後可跟一個可選的 else 語句，else 語句在布林代數為假時執行。

語法

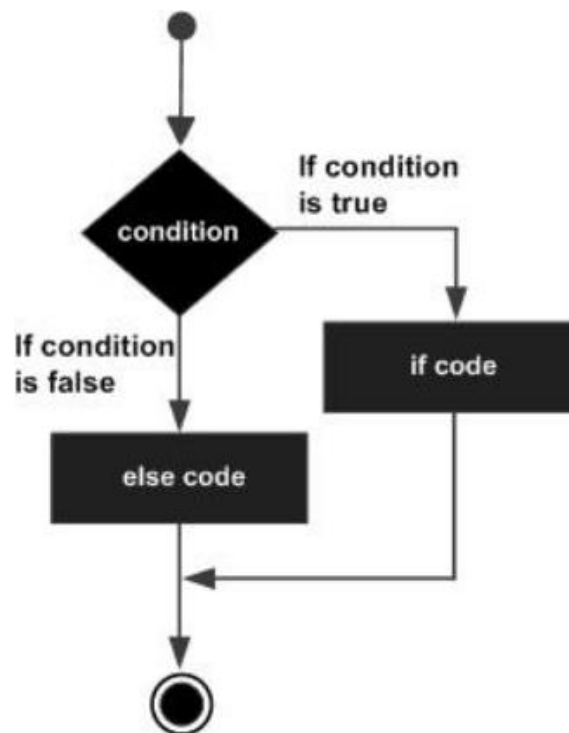
C++ 中 if...else 語句的語法：

```
if(boolean_expression)  
{  
    // 如果布林代數為真將執行的語句  
}  
else  
{  
    // 如果布林代數為假將執行的語句  
}
```

如果布林代數為 true，則執行 if 塊內的程式碼。

如果布林代數為 false，則執行 else 塊內的程式碼。

流程圖



實例 4-02

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <iostream>
4  using namespace std;
5
6  int main ()
7  {
8      int a = 100;
9
10     if( a < 20 ) {
11         cout << "a 小於 20" << endl;
12     }
13     else {
14         cout << "a 大於 20" << endl;
15     }
16     cout << "a 的值是 " << a << endl;
17
18     return 0;
19 }
```

當上面的程式碼被編譯和執行時，它會產生下列結果：

```
a 大於 20  
a 的值是 100
```

if...else if...else 語句

一個 if 語句後可跟一個可選的 else if...else 語句，這可用於測試多種條件。

當使用 if...else if...else 語句時，以下幾點需要注意：

1. 一個 if 後可跟零個或一個 else，
else 必須在所有 else if 之後。
2. 一個 if 後可跟零個或多個 else if，
else if 必須在 else 之前。
3. 一旦某個 else if 匹配成功，
其他的 else if 或 else 將不會被測試。

語法

C++ 中的 if...else if...else 語句的語法：

```
if(boolean_expression 1)  
{  
    // 當布林代數 1 為真時執行  
}  
else if( boolean_expression 2)  
{  
    // 當布林代數 2 為真時執行  
}  
else if( boolean_expression 3)  
{  
    // 當布林代數 3 為真時執行  
}  
else
```

```
{  
    // 當上面條件都不為真時執行  
}
```

實例 4-03

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  #include <iostream>  
4  using namespace std;  
5  
6  int main ()  
7  {  
8      int a = 100;  
9  
10     if( a == 10 ) {  
11         cout << "a 的值是 10" << endl;  
12     }  
13     else if( a == 20 ) {  
14         cout << "a 的值是 20" << endl;  
15     }  
16     else if( a == 30 ) {  
17         cout << "a 的值是 30" << endl;  
18     }  
19     else {  
20         cout << "沒有匹配的值" << endl;  
21     }  
22     cout << "a 的準確值是 " << a << endl;  
23  
24     return 0;  
25 }
```

當上面的程式碼被編譯和執行時，它會產生下列結果：

```
沒有匹配的值  
a 的準確值是 100
```

巢狀 if 語句

在 C++ 中，巢狀 if-else 語句是合法的，這意味著您可以在一個 if 或 else if 語句內使用另一個 if 或 else if 語句。

語法

```
if( boolean_expression 1)
{
    // 當布林代數 1 為真時執行
    if(boolean_expression 2)
    {
        // 當布林代數 2 為真時執行
    }
}
```

您也可以將 else if...else 巢狀，方式與巢狀 if 語句相似。

實例 4-04

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <iostream>
4  using namespace std;
5
6  int main ()
7  {
8      int a = 100;
9      int b = 200;
10
11     if( a == 100 ) {
12         if( b == 200 ) {
13             cout << "a 的值是 100，且 b 的值是 200" << endl;
14         }
15     }
16     cout << "a 的準確值是 " << a << endl;
17     cout << "b 的準確值是 " << b << endl;
18
19     return 0;
20 }
```


當上面的程式碼被編譯和執行時，它會產生下列結果：

```
a 的值是 100，且 b 的值是 200
a 的準確值是 100
b 的準確值是 200
```

switch 語句

一個 switch 語句允許測試一個變數等於多個值時的情況。每個值稱為一個 case，且被測試的變數會對每個 switch case 進行檢查。

語法

```
switch(expression){
    case constant-expression :
        statement(s);
        break; // 可選的
    case constant-expression :
        statement(s);
        break; // 可選的

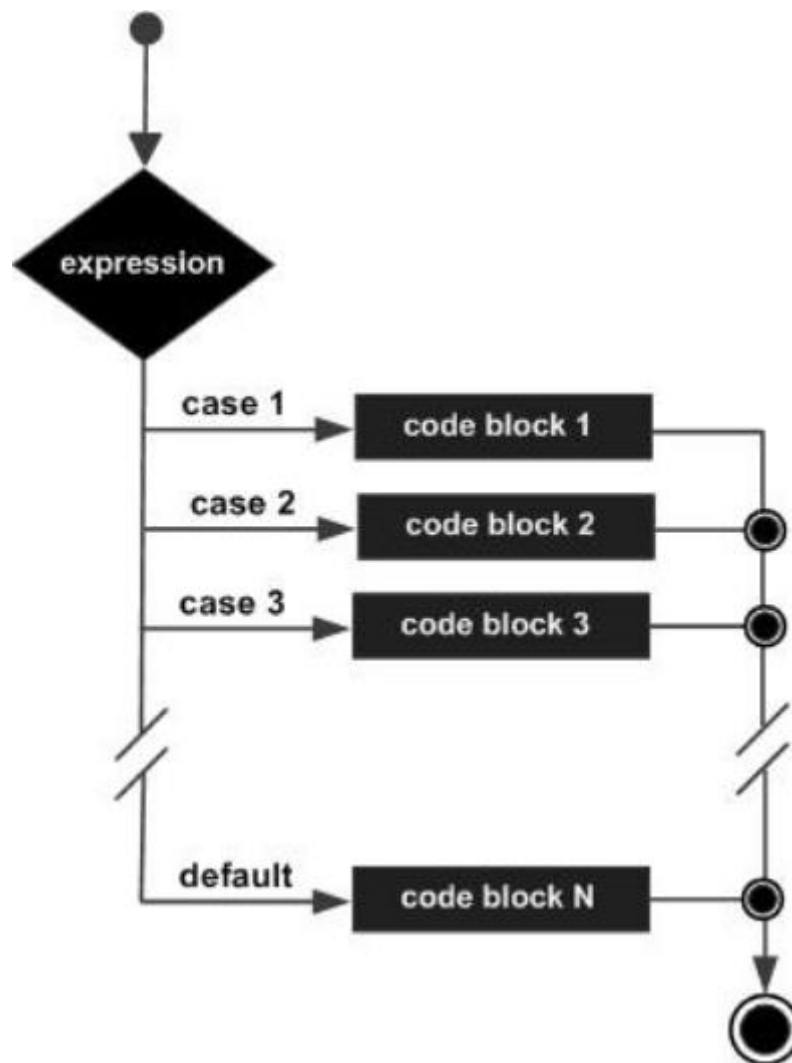
    // 您可以有任意數量的 case 語句
    default : // 可選的
        statement(s);
}
```

switch 語句必須遵循下面的規則：

1. switch 語句中的 expression 必須是一個整數型或枚舉類型，或者是一個 class 類型，其中 class 有一個單一的轉換函數將其轉換為整數型或枚舉類型。
2. 在一個 switch 中可以有任意數量的 case 語句。每個 case 後跟一個要比較的值和一個冒號。
3. case 的 constant-expression 必須與 switch 中的變數具有相同的資料類型，且必須是一個常量或字面量。

4. 當被測試的變數等於 case 中的常量時，case 後跟的語句將被執行，直到遇到 break 語句為止。
5. 當遇到 break 語句時，switch 終止，控制流將跳轉到 switch 語句後的下一行。
6. 不是每一個 case 都需要包含 break。如果 case 語句不包含 break，控制流將會繼續後續的 case，直到遇到 break 為止。
7. 一個 switch 語句可以有一個可選的 default case，出現在 switch 的結尾。default case 可用於在上面所有 case 都不為真時執行一個任務。default case 中的 break 語句不是必需的。

流程圖



實例 4-05

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <iostream>
4  using namespace std;
5
6  int main () {
7      char grade = 'D';
8
9      switch(grade) {
10         case 'A' :
11             cout << "很棒 !" << endl;
12             break;
13         case 'B' :
14         case 'C' :
15             cout << "做得好" << endl;
16             break;
17         case 'D' :
18             cout << "您通過了" << endl;
19             break;
20         case 'F' :
21             cout << "最好再試一下" << endl;
22             break;
23         default :
24             cout << "無效的成績" << endl;
25     }
26     cout << "您的成績是 " << grade << endl;
27
28     return 0;
29 }
```

當上面的程式碼被編譯和執行時，它會產生下列結果：

```
您通過了
您的成績是 D
```

巢狀 switch 語句

您可以把一個 switch 作為一個外部 switch 的語句序列的一部分，即可以在一個 switch 語句內使用另一個 switch 語句。

即使內部和外部 switch 的 case 常量包含共同的值，也沒有矛盾。

C++ 中的 switch 語句允許至少 256 個巢狀層次。

語法

```
switch(ch1) {  
    case 'A':  
        cout << "這個 A 是外部 switch 的一部分";  
        switch(ch2) {  
            case 'A':  
                cout << "這個 A 是內部 switch 的一部分";  
                break;  
            case 'B': // 內部 B case 程式碼  
            }  
            break;  
        case 'B': // 外部 B case 程式碼  
    }  
}
```

實例 4-06

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <iostream>
4  using namespace std;
5
6  int main () {
7      int a = 100;
8      int b = 200;
9
10     switch(a) {
11         case 100:
12             cout << "這是外部 switch 的一部分" << endl;
13             switch(b) {
14                 case 200:
15                     cout << "這是內部 switch 的一部分" << endl;
16             }
17     }
18     cout << "a 的準確值是 " << a << endl;
19     cout << "b 的準確值是 " << b << endl;
20
21     return 0;
22 }
```

當上面的程式碼被編譯和執行時，它會產生下列結果：

```
這是外部 switch 的一部分
這是內部 switch 的一部分
a 的準確值是 100
b 的準確值是 200
```

? : 運算符號

條件運算符號 ? :，可以用來替代 if...else 語句。它的一般形式如下：

```
Exp1 ? Exp2 : Exp3;
```

其中，Exp1、Exp2 和 Exp3 是表達式。**請注意，冒號的使用和位置。**

? 表達式的值是由 Exp1 決定的。

輔仁中學 ACPS C++ 程式課程上課講義

如果 Exp1 為真，則回傳 Exp2 的值，結果即為整個 ? 表達式的值。

如果 Exp1 為假，則回傳 Exp3 的值，結果即為整個 ? 表達式的值。