4.1 selection criteria
4.2 the if-else statement
4.3 nested if statements
4.4 the switch statement
4.5 a case study: solving quadratic equations
4.6 a closer look: program testing
4.7 common programming errors
4.8 chapter summary
Objectives

In this chapter, you will learn about:
- Selection criteria 選擇原則
- The if-else statement 選擇敘述
- Nested if statements 巢狀if敘述
- The switch statement 另一種選擇敘述
- Program testing 程式測試
- Common programming errors 常見程式錯誤
Selection Criteria 選擇原則

- **if-else** statement: Implements a decision structure for two alternatives 兩個互斥選項

Syntax: 語法

```cpp
if (condition)
    statement executed if condition is true;
else
    statement executed if condition is false;
```
Selection Criteria (continued)

- The condition is evaluated to its numerical value: 条件评估结果会以数值方式表示
  - A non-zero value is considered to be true 非零为真
  - A zero value is considered to be false 零为假
- The else portion is optional; it is executed only if the condition is false 放在else的部份可有可无(必要)
- The condition may be any valid C++ expression 条件可以是任何C++叙述
Relational Operators 關連運算符號

- **Relational expression**: Compares two operands or expressions using relational operators 關係運算式

<table>
<thead>
<tr>
<th>Relational Operator</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;</code></td>
<td>Less than</td>
<td>age &lt; 30</td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>Greater than</td>
<td>height &gt; 6.2</td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>Less than or equal to</td>
<td>taxable &lt;= 20000</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>Greater than or equal to</td>
<td>temp &gt;= 98.6</td>
</tr>
<tr>
<td><code>==</code></td>
<td>Equal to</td>
<td>grade == 100</td>
</tr>
<tr>
<td><code>!=</code></td>
<td>Not equal to</td>
<td>number != 250</td>
</tr>
</tbody>
</table>

**Table 4.1 C++’s Relational Operators 關係運算符號**
Relational Operators (continued)

• Relational expressions are evaluated to a numerical value of 1 or 0 only: 關係運算式的值是1或0
  – If the value is 1, the expression is true 若1是真
  – If the value is 0, the expression is false 若0是假
• char values are automatically coerced to int values for comparison purposes 字元資料自動轉換成整數再加以比較
• Strings are compared on a character by character basis 字串使用一個一個字母比對
  – The string with the first lower character is considered smaller 第一個較前字母的字串為小
Relational Operators (continued)

- Examples of string comparisons

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Interpretation</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Hello&quot; &gt; &quot;Good-bye&quot;</td>
<td>1</td>
<td>true</td>
<td>The first H in Hello is greater than the first G in Good-bye.</td>
</tr>
<tr>
<td>&quot;SMITH&quot; &gt; &quot;JONES&quot;</td>
<td>1</td>
<td>true</td>
<td>The first S in SMITH is greater than the first J in JONES.</td>
</tr>
<tr>
<td>&quot;123&quot; &gt; &quot;1227&quot;</td>
<td>1</td>
<td>true</td>
<td>The third character in 123, the 3, is greater than the third character in 1227, the 2.</td>
</tr>
<tr>
<td>&quot;Behop&quot; &gt; &quot;Beehive&quot;</td>
<td>1</td>
<td>true</td>
<td>The third character in Behop, the h, is greater than the third character in Beehive, the second e.</td>
</tr>
<tr>
<td>&quot;He&quot; == &quot;She&quot;</td>
<td>0</td>
<td>false</td>
<td>The first H in He is not equal to the first S in She.</td>
</tr>
<tr>
<td>&quot;plant&quot; &lt; &quot;planet&quot;</td>
<td>0</td>
<td>false</td>
<td>The t in plant is greater than the e in planet.</td>
</tr>
</tbody>
</table>
Logical Operators 邏輯運算符號

- **AND (&&)**: Condition is true only if both expressions are true 而且(兩者同時為真方為真)
- **OR (||)**: Condition is true if either one or both of the expressions is true 或(有一者為真即為真)
- **NOT (!)**: Changes an expression to its opposite state; true becomes false, false becomes true 非(真假互換)
Logical Operators (continued)

Table 4.2 Operator Precedence and Associativity 運算優先順序

<table>
<thead>
<tr>
<th>Operator</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>! unary - ++ --</td>
<td>Right to left</td>
</tr>
<tr>
<td>* / %</td>
<td>Left to right</td>
</tr>
<tr>
<td>+ -</td>
<td>Left to right</td>
</tr>
<tr>
<td>&lt; &lt;= &gt; &gt;=</td>
<td>Left to right</td>
</tr>
<tr>
<td>== !=</td>
<td>Left to right</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>Left to right</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>+= -= *= /=</td>
<td>Right to left</td>
</tr>
</tbody>
</table>
A Numerical Accuracy Problem 數值精確度問題

• Comparing single and double precision values for equality (==) can lead to errors because values are stored in binary 因為數值以二進位儲存，所以比較單精度和雙精度數值是否相等可能造成錯誤
  – Example:  
    \[ \text{abs(operandOne - operandTwo)} < 0.000001 \]
The *if-else* Statement

- *if-else* performs instructions based on the result of a comparison 此一敘述的執行視比較的結果而定
- Place statements on separate lines for readability 放在不同行只是為了易讀
- Syntax: 語法

```cpp
if (expression)  // no semicolon here
    statement1;
else               // no semicolon here
    statement2;
```

C++ for Engineers and Scientists, Third Edition
The if-else Statement (continued)

Figure 4.2
The if-else flowchart 流程圖
The if-else Statement (continued)

Program 4.1

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    double radius;
    cout << "Please type in the radius: ";
    cin >> radius;

    if (radius < 0.0)
        cout << "A negative radius is invalid" << endl;
    else

        cout << "The area of this circle is " << 3.1416 * pow(radius,2) << endl;
    return 0;
}
```
Compound Statement: A sequence of single statements contained between braces 一系列的單獨敘述的組合
- Creates a block of statements 用括號產生一個區塊的敘述
- Block of statements can be used anywhere that a single statement is legal 區塊敘述可以放在任何單一敘述的地方
- Any variable declared within a block is usable only within that block 在一個區塊中宣告的變數只在區塊中有效
- Scope: The area within a program where a variable can be used 變數有效範圍稱Scope
  - A variable’s scope is based on where the variable is declared 變數的有效範圍視宣告的地方而定
Block Scope 區塊範圍

- Statements contained in compound statement are a single block of code
- **Scope of the variable:** Area in a program where a variable can be used 變數有效範圍
Block Scope (continued)

```c++
{  // start of outer block
    int a = 25;
    int b = 17;

    cout << "The value of a is " << a
         << " and b is " << b << endl;

    {  // start of inner block
        double a = 46.25;

        int c = 10;
        cout << "a is now " << a
             << " b is now " << b
             << " and c is " << c << endl;
    }  // end of inner block

    cout << "a is now " << a
         << " and b is " << b << endl;

}  // end of outer block
```

C++ for Engineers and Scientists, Third Edition
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    char tempType;
    double temp, fahrenheit, celsius;

    cout << "Enter the temperature to be converted: ";
    cin >> temp;
    cout << "Enter an f if the temperature is in Fahrenheit ";
    cout << "\n or a c if the temperature is in Celsius: ";
    cin >> tempType;

    cout << setiosflags(ios::fixed)
          << setiosflags(ios::showpoint)
          << setprecision(2);

    if (tempType == 'f')
    {
        celsius = (5.0 / 9.0) * (temp - 32.0);
        cout << "\nThe equivalent Celsius temperature is "
             << celsius << endl;
    }
    else
    {
        fahrenheit = (9.0 / 5.0) * temp + 32.0;
        cout << "\nThe equivalent Fahrenheit temperature is "
             << fahrenheit << endl;
    }
    return 0;
}
One-Way Selection 單通路選擇

- **One-way selection**: An if statement without the optional else portion 沒有else的部分的選擇敘述

![Flowchart of a one-way selection if statement](image)

**Figure 4.3** A one-way selection if statement
Problems Associated with the if-else Statement 問題

- Common problems with if-else statements: 常見問題
  - Misunderstanding what an expression is 不了解何為表示式
  - Using the assignment operator (=) instead of the relational operator (==) 使用兩個等號做比較
Nested if Statements 巢狀if敘述

- if-else statement can contain any valid C++ statement, including another if-else 選擇敘述中可以有另一個選擇敘述
- Nested if statement: an if-else statement completely contained within another if-else 巢狀敘述
- Use braces to block code, especially when inner if statement does not have its own else 使用括號來區分(尤其當內部的if沒有else)
Figure 4.4a
Nested within the if part
The if-else Chain 選擇鏈

- **if-else** chain: A nested if statement occurring in the else clause of the outer if-else 内部if敘述放在外
  部if敘述的else部分

- If any condition is true, the corresponding statement is executed and the chain terminates 如果任一條件為真，
  則相關的敘述執行且選擇鏈終止

- Final else is only executed if no conditions were true 最後的else只在全部條件為假時執行
  
  - Serves as a catch-all case

- if-else chain provides one selection from many possible alternatives 選擇鏈提供多方案中的單一選擇
The if-else Chain (continued)

Figure 4.4b
Nested within the else part 在else的部份套入其他選擇
The if-else Chain (continued)

- General form of an if-else chain 選擇鏈一般形式

```cpp
if (expression_1)
    statement1;
else if (expression_2)
    statement2;
else if (expression_3)
    statement3;
    ...
else if (expression_n)
    statement_n;
else
    last_statement;
```
The switch Statement

- **switch** statement: Provides for one selection from many alternatives 提供多方案中的單一選擇
- **switch** keyword starts the statement 關鍵字是switch
  - Is followed by the expression to be evaluated 其後跟著要計算的表示式
- **case** keyword identifies a value to be compared to the switch expression 用case關鍵字帶領比對的值
  - When a match is found, statements in this case block are executed 有對應值時相關敘述執行
- All further cases after a match is found are executed unless a **break** statement is found 除非有break否則後面的cases也會執行
The `switch` Statement (continued)

- `default` case is executed if no other case value matches were found 預設選項(無符合數值時執行)
- `default` case is optional 非必要
#include <iostream>

using namespace std;

int main()
{
    int opselect;
    double fnum, snum;

    cout << "Please type in two numbers: ";
    cin >> fnum >> snum;
    cout << "Enter a select code: ";
    cout << "\n  1 for addition";
    cout << "\n  2 for multiplication";
    cout << "\n  3 for division : ";
    cin >> opselect;

    switch (opselect)
    {
    case 1:
        cout << "The sum of the numbers entered is " << fnum+snum;
        break;
    case 2:
        cout << "The product of the numbers entered is " << fnum*snum;
        break;
    case 3:
        cout << "The first number divided by the second is " << fnum/snum;
        break;
    }

    cout << endl;
    return 0;
}
A Case Study: Solving Quadratic Equations

- **Data validation**: Use defensive programming techniques to validate user input. Includes code to check for improper data before an attempt is made to process it further.

- **Solving quadratic equations**: Use the software development procedure to solve for the roots of a quadratic equation.
A Closer Look: Program Testing 程式測試

- Theory: A comprehensive set of test runs would test all combinations of input and computations, and would reveal all errors 一套完整的測試考慮所有輸入和計算的組合可以顯示所有錯誤

- Reality: There are too many combinations to test for any program except a very simple one 但事實上要測試的案例太多

- Example:
  - One program with 10 modules, each with five if statements, always called in the same order
  - There are $2^5$ paths through each module, and more than $2^{50}$ paths through the program!

C++ for Engineers and Scientists, Third Edition
• Conclusion: there is no error-free program, only one in which no errors have recently been encountered 沒有完全無錯的程式，只有尚未找出錯誤的程式
Common Programming Errors

- Using the assignment operator (\(=\)) instead of the relational operator (\(==\)) for an equality test
- Placing a semicolon immediately after the condition 條件之後沒有分號
- Assuming a structural problem with an if-else causes the error instead of focusing on the data value being tested
- Using nested if statements without braces to define the structure
Summary

• Relational expressions, or conditions, are used to compare operands
• If the relation expression is true, its value is 1; if false, its value is 0
• Use logical operators && (AND), || (OR), and ! (NOT) to construct complex conditions
• if-else allows selection between two alternatives
Summary (continued)

- An `if` expression that evaluates to 0 is false; if non-zero, it is true
- `if` statements can be nested
- Chained `if` statement provides a multiway selection
- Compound statement: contains any number of individual statements enclosed in braces
Summary (continued)

- **switch statement**: Provides a multiway selection
- **switch expression**: Evaluated and compared to each case value
  - If a match is found, execution begins at that case’s statements and continues unless a break is encountered