

頁碼	位置	原文	修正																					
27	第 8 行	反覆敘述 (Iteration Statement)：讓特定程式碼可以反覆執行的敘述。	重複敘述 (Repetition Statement，又稱為迭代敘述 Iteration Statement)：讓特定程式碼可以重複執行的敘述。																					
27	第 18 行	跳躍與反覆	跳躍與重複																					
51-52	BMI-2.cpp 的執行結果	此程式範例的執行結果中，所輸入的體重 66.5 有誤	應改為 65.5																					
119	第 1 個及第 3 個程式碼區塊的第 3 行	cout << "b1=" << b << ...	cout << "b1=" << b1 << ...																					
126	習題 2	關於 C 語言	關於 C++ 語言																					
199	習題 7 的第 3 個執行結果方塊的最後一行	57r+0.1.	57e+01.																					
200	習題 10	請題的執行結果可參考如下：	本題的執行結果可參考如下：																					
200	習題 10 的第 1 個執行結果的第 2 行	***xyz*** ↓	***xyz***** ↓																					
201	習題 10 的第 2 個執行結果的第 2 行	***xyz*** ↓	*xyz* ↓																					
201	習題 10 的第 3 個執行結果的第 2 行	***xyz*** ↓	xyz ↓																					
203	第 8 行	線性執行動線就是順序結構	線性執行動線就是循序結構																					
203	第 14 行	兩個常用的條件敘述	兩個常用的選擇敘述																					
204	倒數第 1 行	下列的邏輯運算式示範了	下面的範例程式示範了																					
206	表 6-3	<table><tr><th>符號</th><th>範例</th><th>意義</th></tr><tr><td>==</td><td>a == b</td><td>a 是否等於 b</td></tr><tr><td>!=</td><td>a != b</td><td>a 是否不等於 b</td></tr></table>	符號	範例	意義	==	a == b	a 是否等於 b	!=	a != b	a 是否不等於 b	<table><tr><th>符號</th><th>意義</th><th>一元或二元運算子</th></tr><tr><td>&&</td><td>AND</td><td>二元</td></tr><tr><td> </td><td>OR</td><td>二元</td></tr><tr><td>!</td><td>NOT</td><td>一元</td></tr></table>	符號	意義	一元或二元運算子	&&	AND	二元		OR	二元	!	NOT	一元
符號	範例	意義																						
==	a == b	a 是否等於 b																						
!=	a != b	a 是否不等於 b																						
符號	意義	一元或二元運算子																						
&&	AND	二元																						
	OR	二元																						
!	NOT	一元																						
207	第 11 行	例如 quit 是一個整數變數	例如 quit 是一個 bool 型態變數																					
207	倒數第 4 與第 6 行	變數 x 是否的值介於 a 與 b 之間	變數 x 的值是否介於 a 與 b 之間																					
207	倒數第 2 行	<x) < b 意即	<x) < b, 意即																					
207	倒數第 1 行	我們想要以邏輯運算式	我們想要以邏輯運算式																					
208	第 1 行	的值介於 a 與 b 之間	(刪除的值) 介於 a 與 b 之間																					
226	圖 6-3 標題	Example 6-3 的 fareSwitch.cpp 程式流程圖	Example 6-4 的 fareSwitch.cpp 程式流程圖																					

227	習題 3	<code>cout << 'A'+(x<0?0:x);</code>	<code>cout << char('A'+(x<0?0:x));</code>
235	第 1 行末與倒數第 2 行	重複	重複
239	第 1 行及倒數第 2 行	重複	重複
241	第 10 行	再以 <code>taskkill /PID -t</code> 指令將程式從系統移除。	再以 <code>taskkill /PID #####-t</code> 指令將程式從系統移除，其中#####為程式的 PID。
247	第 1 行	以下是一些 <code>do while</code> 迴圈的應用範例	以下是一些 <code>for</code> 迴圈的應用範例
252	最後 1 行末	<code>;i++</code> 嗎？	<code>;i++)</code> 嗎？
255	倒數第 8 行	將 Example 7-12 的 <code>sumUserInput.cpp</code> 加以改寫	將 Example 7-13 的 <code>sumUserInput.cpp</code> 加以改寫
256	第 1 行末	而是省略當次執行時未完成的程式碼	而是省略當次執行迴圈主體時未完成的程式敘述
259	習題 2	2. 以下關於 <code>break</code> 與 <code>continue</code> 的說明何者正確？	2. 以下關於 <code>break</code> 與 <code>continue</code> 的說明何者正確？(複選)
269	第 2 行	，因為其後的 <code>?</code> 代表它可以使用 0 次或 1 次)	(因為其後的 <code>?</code> 代表它可以使用 0 次或 1 次)，
276	第 14 行	<code>islower(int c)</code>	<code>int islower(int c)</code>
277	Example 8-3 程式檔案名稱	<code>bubberSort.cpp</code>	<code>bubbleSort.cpp</code>
309	第 14 行末	傳遞給參數 <code>p</code> 的接收	傳遞給參數 <code>p</code> 接收
334	第 1 行	C 語言允許	C++ 語言允許
346	Example 10-14 標題	使用指標走訪操作二維陣列	二維陣列走訪與操作範例
356	習題 1	1. 請參考以下的程式碼片段：	1. 請參考以下的程式碼片段：
358	習題 6	<pre>int data[3][5] = {{ 1, 2, 3, 4, 5}, { 6, 7, 8, 9,10}, {11,12,13,14,15}}; int (*p)[5] = &data[0]; cin >> col; for(;p<=&data[2];p++) { cout << (a) [col] << endl; }</pre>	<pre>int data[3][5] = {{ 1, 2, 3, 4, 5},^{e1} { 6, 7, 8, 9,10},^{e2} {11,12,13,14,15}};^{e3} int (*p)[5] = &data[0];^{e4} cin >> col;^{e5} for(;p<=&data[2];p++)^{e6} {^{e7} cout << (a) [col] << endl;^{e8} }^{e9}</pre>
367	第 10 行末	後續再就命令列引數以及 C++ 語言	後續再就 C++ 語言
367	第 11 行	新增的 <code>string</code> 類別，分別加以說明	的 <code>string</code> 類別加以說明
369	圖 11-2(a)		
388	倒數第 1 行	本節僅列示範一些常用的字串處理函式：	本節僅列示一些常用的字串處理函式：
409	第 7 行	其中包含產名代碼、名稱、單價	其中包含產品代碼、名稱、單價
411	倒數第 4 行	下載取得範例程式檔案	驗證結構體記憶體空間
473	下方的程式碼	<code>struct Student</code>	<code>class Student</code>

480	習題 2	<pre> class Customer { public: string name; string ID; string accountNo; int balance; } class Student { public: string bob_name; string bob_SID; int bob_score; bool isPass() { return (score>=60); } } </pre>	<pre> class Customer { public: string name; string ID; string accountNo; int balance; } class Student { public: string name; string SID; int score; bool isPass() { return (score>=60); } } </pre>
500	第 5 行末	>showInfo();。	>showInfo();。當不再需要使用動態配置的物件實體時，可以使用 delete 將該配置的空間加以回收，例如 delete bob;。
518	習題 6	<pre> 1 using namespace std; 2 #include <iostream> 3 4 class Person 5 { 6 public: 7 string name; 8 void printName() 9 { 10 cout << "Name: " << name << endl; 11 } 12 }; 13 </pre>	<pre> 1 #include <iostream> 2 using namespace std; 3 4 class Person 5 { 6 public: 7 string name; 8 void printName() 9 { 10 cout << "Name: " << name << endl; 11 } 12 }; 13 </pre>
546	第 4 行	並在第 22 行	並在第 20 行
604	第 3 行	本附錄所要介紹	本節所要介紹
622	第 3 行	定義了三次	定義了兩次
628	倒數第 5 行	以及在第 24 行定義並在第 27-32 行，我們將兩個函式寫在類別內部，並將一個函式實作於類別之外。	我們將兩個函式寫在類別內部，並在第 24 行定義、在第 27-32 行將一個函式實作於類別之外。
629	倒數第 2 行	只要將 + 運算字多載函式的傳回值與 << 運算字多載函式	只要將 + 運算子多載函式的傳回值與 << 運算子多載函式
654	第 5 行	等同於強制要求子類別一定要覆寫父類別的虛擬函式	等同於強制要求檢查子類別所覆寫的是否是父類別的虛擬函式
656	習題 3	<pre> public: void setNationality(string n) { nationality=n; } string getNationality() { return nationality; } void showInfo(){ cout << "ForeignStudent::showInfo() " << getName() << " Score=" << getScore() << "[" << getNationality() << "]" << endl; } }; </pre>	<pre> public: void setNationality(string n) { nationality=n; } string getNationality() { return nationality; } void showInfo() { cout << "ForeignStudent::showInfo() " << getName() << " Score=" << getScore() << "[" << getNationality() << "]" << endl; } }; </pre>