|  |  |  |  |
| --- | --- | --- | --- |
| 頁數 | 行數 | 修正 |  |
| 15 | 15 | （的確，有些OOP的純粹主義者認為在C中加入OOP特性無異是在豬身上安裝翅膀，雖然是有效率的豬） | 稿件上註明「不順？」，但未指示如何修改。 |
| 22 | 3、4、6、7 | CC my.c precious.C  my.C，  CC my.c precious.o  my.C | 稿件上註明「c的大小寫」，但未指示如何修改。 |
| 23 | -7 | 、Execute、Run及debug（但不一定所有的選擇都會在相同的IDE中！）。 | 稿件上註明「d要大寫嗎？」，但未指示如何修改。 |
| 33 | 8 | 實際上，C99標準已經在C語言中加入//註解。 | 稿件上在「C99」畫底線，但未指示如何修改。 |
| 34 | -5 |  | 稿件上註明「c小寫？」，但未指示如何修改。 |
| 45 | -9、-10 | 但在這程式您將要加入兩行  cin.get()敘述，使程式的輸出結果，可以順利的顯示於螢幕。為什麼要兩行cin.get()呢？ |  |
| 70 | -10 | long long n\_llong = LLONG\_MAX; |  |
| 71 | 7 | 若使用的系統沒有支援long long，請將程式中有使用long long的敘述刪除。 |  |
| 74 | 7 | int emus{7}; // set emus to 7 |  |
| 105 | 9 | 以c4的初始化來說，我們知道x的值為66，但對於編譯器來說，x是個變數，可能會帶有其他更大的數值。編譯器並不負責追蹤x變數初始化後，到被用來初始化c4之間，x所帶的數值會如何變動。 |  |
| 121 | 12 | ，它會顯示前7個字元， |  |
| 144 | 10 | inflatable duck = {"Daphne", 0.12. 9.98} | 稿件上註明「需要分號嗎？」，但未指示如何修改。 |
| 146 | 19~20 |  |  |
| 147 | -20~-17、-13 |  |  |
| 150 | -5 | 做兩件事情： |  |
| 159 | -10 | 產生儲存資料的位址是不同的步驟， |  |
| 160 | -15 | 型態int值的位址不代表pt本身是int型態。 |  |
| 175 | -8 | 使用const表示您可用bird存取字串 |  |
| 187 | -14 | ，也是std namespace的一部分。 | 稿件上在「std namespace」畫底線，但未指示如何修改。 |
| 209 | 8 | 因此，C++保證副作用（遞增guests）會在程式執行至cout以前發生。 |  |
| 301 | 2 | 撰寫一程式讀入鍵盤輸入，直到碰到@符號，並重複顯示輸入，但數字例外， | 稿件上在「並重複顯示輸入，但數字例外」畫底線，但未指示如何修改。 |
| 320 | 14 | 所以將它寫成函數是合理的，您就不必每次計算陣列和就得寫一次迴圈。 | 稿件上在「您就不必每次計算陣列和就得寫一次迴圈」畫了問號，但未指示如何修改。 |
| 334 | 5~6 | 關鍵字const和指標有兩種不同的用法。第一個方法是使指標指向不變的物件，這可避免用指標改變所指之值。第二個方法是使指標本身為常數，避免改變指標所指的位置。 | 未指示如何修改。 |
| 335 | -9~-10 | 您可將const資料或是非const資料的位址指定給指向const的指標，前提是資料的型態不是指標，但您可將非const資料的位址指定給非const指標。 | 這三句畫了問號，但未指示如何修改。 |
| 343 | 4 | ，就會釋放pstr所佔的記憶體（並不是字串的記憶體）。 |  |
| 357 | 1~2 | EXPENSES  Spring: $212 |  |
| 409 | 7 | 如果資料物件是適當大小的結構，則使用const指標 |  |
| 444 | -15 | 給v1、v2、v3、v4，以及v5。 |  |
| 459 | -14 | 留下的register關鍵字用來顯示辨別自動變數。 |  |
| 462 | 10、17、21~22 | 靜態變數能以0做初始化，可能經由常數運算式初始化…  以零做初始化，以及常數運算式初始化…  如果變數使用常數運算式初始化，而編譯器可單獨從檔案內容（包括匯入的標頭檔）裡解析該運算式，那麼便可以執行常數運算式初始化。 |  |
| 463 | 1、4 | 常數表達式並不限定於使用字面常數的算術運算式。  用來擴充建立常數運算式的方式； |  |
| 464 | 圖9.4 | 此檔案宣告變數process\_status，編譯器將為其配置記憶體空間。  此檔案使用extern，指示程式參考定義於其他檔案的process\_status變數。 |  |
| 467 | 6~7 | 現在您要選擇使用全域或區域變數，您應該使用何者呢？首先，全域變數較有魅力，因為所有函數均可存取全域變數，省去了傳遞引數的麻煩。但是容易存取卻要付出很高的代價，因為它屬於不可靠的程式。 |  |
| 471 | 12 |  |  |
| 481 | -7 | 第二件要注意的是，在第二次呼叫一般使用的new時，會讓new去找尋一塊新  的記憶體空間 ── 由0006E4AB0開始的空間。 |  |
| 485 | 圖9.6 |  |  |
| 523 | -12 | 換言之，Retort()不只是型態char\*的成員函數， |  |
| 537 | -4 | void Stock::show() const // promises not to change invoking object |  |
| 611 | 範例程式11.18 | Stonewt pavarotti = 275  改為  Stonewt incognito = 275 |  |
| 639 | 10 | 複製建構函數是將物件複製到最新產生的物件， |  |
| 640 | -9 | 回到StringBad：複製建構函數何處出錯 |  |
| 644 | 5  -3 | 更多StringBad 的問題：指定運算子  和複製建構函數一樣，指定運算子的隱含式做法是… |  |
| 648 | -18 | C++11 null指標 |  |
| 694 | 10 | 每個週期表示1分鐘，期間程式會做下列事項： |  |