前言

本書是專門討論實用的 C++ 程式設計，不只涉及語言的運用技巧，也會說明寫作風格與除錯方法。內容討論到程式的整個生命週期，包含了觀念、設計、編寫、除錯、發佈、文件化、維護及修訂等。

風格是本書特別強調的重點。寫一個好的程式不只是鍵入程式碼而已，必須要將寫作與程式設計技巧，密切結合成為一個藝術作品。優良的程式不僅是運作正常，而且應簡單易懂。註解讓程式設計師在程式中包含說明的文字，編寫清晰、註解良好的程式極受推崇。

程式應盡可能簡單，避免使用「偷吃步」。小聰明與複雜性會使程式變得一無是處。本書強調簡單、實用的規則。例如，在 C++ 中的 15 個運算子優先順序，可以簡化成二個規則：

1. 先乘除後加減。
2. 其餘東西都用括號括住。

試想有二個程式，其中一個是由絕頂聰明的程式設計師所寫的，裡面用了所有的小技巧。該程式沒有任何註解，但是可以運作。另一個程式加了很清楚的註解而且結構良好，但是無法運作。哪個程式比較有用？從長遠的角度來看，無法運作的程式較有用，因為它可以容易地修改和維護。雖然用小技巧的程式目前運作正常，但是遲早需要修改。修改使用小技巧的程式將是最困難的工作。

本書適用範圍

本書寫給沒有任何程式設計經驗的人、了解 C 語言並想提昇技術到 C++ 的程式設計師；以及那些了解 C++ 但要改善其程式設計風格與可靠度的人。你必須在電腦實際操作，並知道如何使用像是文字編輯器及檔案系統的基本功能。

學習電腦語言最好的方式是寫程式與除錯。凌晨兩點還在琢磨一個有問題的程式，只為了找出在應該鍵入 == 的地方卻只輸入 =，這是一種非常有效的教學手段。本書包含許多常見程式設計的錯誤範例（它們在書中被標示為錯誤程式）。我鼓勵你把程式輸入到電腦，加以執行與除錯。這個過程使用簡短的程式介紹你一些常見的錯誤，以便學到如何在自己的程式中去找出及修改這些錯誤（本章最後有如何取得本書範例程式之說明）。

底下的 C++ 編譯器對 C++ 做了一些擴展和修改：

- " 一般 "UNIX 編譯器，可在大多數的 UNIX 系統上運作。
- GNU C++ 編譯器，名稱是 g++(在大多數的 UNIX 系統上都可使用)【註】。
- 可在 MS-DOS/Windows 使用的 Borland C++。
- 可在 MS-DOS/Windows 使用的 Microsoft Visual C++。

雖然主要是用標準 C++ 來解說，但是上面各個版本的編譯器只有些許的不同。本書將清楚說明不同的編譯器對程式設計師所產生的影響，當使用各種編譯器時，將特別指示如何編寫與執行。本書也會介紹使用程式設計工具 *make* 的範例，來自動產生程式。

本書架構

在學會走之前應先學會爬。在第一部分中，將學到如何爬行。這些章節足以教會你編寫非常簡單的程式，從程式設計的結構與風格開始。接著，學習如何使用變數與簡單的判斷和控制敘述。

此時你會學到如何建立最基本的程式；因此第七章〈程式開發的過程〉中，你會了解到整個設計的過程，知道實際程式建立的方法。

第一章〈什麼是 C++〉。告訴你 C++ 的整個觀念，說明它的歷史及用法，並且解說這個語言的組織方式。

第二章〈程式設計基礎〉。說明基本程式設計的過程，以及提供足夠的資訊以便寫一個簡單的程式。

第三章〈風格的展現〉。討論程式設計的風格。介紹如何給程式加註解以及如何寫清楚易懂的程式碼。

註　GNU g++ 編譯器可在 *http://www.gnu.org* 網站上，或聯絡 Free Software Foundation, Inc. 取得，地址電話如後：675 Massachusetts Abenue, Cambridge, MA 02139. (617)876-3296。

第四章〈基本宣告及運算式〉。說明簡單的 C++ 敘述，包含基本變數及**賦值敘述**（**assignment statement**）的細節，以及算術運算子：+、-、*、/ 與 %。

第五章〈陣列、修飾子以及讀取數值〉。包含了陣列及其他較複雜的變數型態，說明縮寫運算子如 ++、--、*=、=、+=、-= 與 %=。

第六章〈判斷及控制敘述〉。說明簡單的判斷敘述，含有 **if**、**else** 及 **for**。另外討論 '=' 與 '==' 二者的問題。

第七章〈程式開發的過程〉。用寫一個簡單的程式來說明需要的步驟，從制定規格到發布。討論的內容有快速原形設計及除錯。

第二部分〈程式設計基礎〉，描述程式設計將用得到的其他簡單敘述及運算子。也會學到如何組織這些敘述到簡單函式。

第八章〈更多控制敘述〉。說明其他的控制敘述，包含 **while**、**break** 及 **continue**，也將討論 **switch** 敘述的細節部分。

第九章〈變數範圍與函式〉。說明區域變數、名稱空間、函式及參數。

第十章〈C++ 前置處理器〉。說明 C++ 前置處理器，可在設計程式時具有很大的彈性，相對地也有很多方式讓你犯錯。並提供簡單的規則避免前置處理器發生錯誤問題。

第十一章〈位元運算〉。說明 C++ 的邏輯運算子如何處理位元資料。

第三部分〈進階的型態及類別〉，你將學習建立高等型態如結構、聯合及類別等所需要的基本宣告及敘述。你也會了解到指標的概念。

第十二章〈進階的型態〉。說明結構及其他進階型態，像是 **sizeof** 運算子及 **enum** 型態。

第十三章〈類別簡述〉。說明**類別**（class）的基本概念。這是 C++ 眾多強大特性中的一種。類別允許你將資料與可在這些資料上的運算，組成到一個物件。

第十四章〈類別再述〉。說明可用類別處理的其他運作。

第十五章〈指標基礎〉。說明 C++ 指標變數與其使用方法。

在第四部分〈進階程式設計的概念〉中，探索高階的程式設計技術。在這個部分將介紹很多 C++ 特性，以便建立複雜但是容易使用的物件或類別。

第十六章〈檔案輸入／輸出〉。說明經過緩衝及未緩衝處理的輸入／輸出（I/O）。討論 ASCII 及二進位檔案的處理，並且說明如何建立一個簡單的檔案。也會說明傳統 C 風格的 I/O 運作。

第十七章〈除錯及最佳化〉。描述如何為程式除錯。以及如何使用互動式除錯程式。不但說明如何對程式除錯，同時也討論如何寫一個容易除錯的程式。這一章內容還說明許多最佳化的技術，使程式執行得更快、更有效率。

第十八章〈運算子重載〉。說明 C++ 允許透過為語言的運算子定義附加的含意來擴充語言。在本章中，將建立一個複雜型態與在該型態上處理的運算子。

第十九章〈浮點運算〉。以一個簡單的十進位浮點運算子的格式，來說明使用浮點運算子與生俱來的問題，像是四捨五入的誤差、損失精確度、overflow 及 underflow【編註】。

第二十章〈進階指標〉。描述指標的高階用法來建立動態結構，像是連結串列（linked list）及樹。

第二十一章〈進階類別〉。說明如何從簡單、基底類別來建立複雜、衍生的類別。

第五部分〈其他語言特性〉裡說明一些其他特性。

第二十二章〈異常〉。說明如何處理程式中意外的情形。

第二十三章〈模組化程式設計〉。指出如何將程式分成多個檔案及使用模組化程式設計技術，將詳細解釋公用程式 make。

第二十四章〈模板〉。允許你定義通用函式或類別，來產生一系列的函式。

第二十五章〈標準模板函式庫〉。描述伴隨 C++ 而來的模板函式庫。這個函式庫由許多**容器模板（container template）**以及相關資料建構而成，它使你以些微的工作就能建立非常複雜和強大的資料結構。

第二十六章〈設計程式〉。討論用於設計程式的一些方法學，比如結構化的程式設計和物件導向設計。不僅討論了設計方法，還說明設計程式的原理。

第二十七章〈整合〉。介紹一個複雜程式從開始構思到完成之間所需的每個步驟細節，特別強調資訊隱藏和模組化程式設計技術，以及物件導向程式設計。

第二十八章〈從 C 到 C++〉。說明如何將 C 程式轉成 C++ 程式，以及說明 C 程式裡會影響到 C++ 程式設計師的潛在陷阱。

第二十九章〈C++ 被遺忘的角落〉。描述了很少使用的 **do/while** 敘述、逗號運算子及 **?:** 運算子。

編註　有譯成上溢、下溢或超上限、超下限。在浮點運算時，overflow 意指數字大到無法表現；underflow 指數字小到無法表現。

第三十章〈程式設計的箴言〉。列出可幫助你寫出好程式的程式設計箴言。

第六部分〈附錄〉。包含額外的 C++ 參考資訊。

附錄 A〈ASCII 表〉。字元碼及其值的列表。

附錄 B〈變數範圍〉。列出一些 C++ 變數型態的數值範圍。

附錄 C〈運算子的優先順序〉。列出決定運算子處理的先後順序規則。

附錄 D〈使用乘冪級數來計算 sine 函數值〉。包含一個電腦如何計算 sine 函數值的程式。

附錄 E〈資源〉。列出本書所提及的程式設計資源資訊。

如果具備 C 語言知識該如何閱讀本書

C++ 是以 C 語言為基礎，如果你瞭解 C 語言，你將會發現第二章到第十二章的內容都很熟悉。

C++ 也在 C 的基礎上引入了大量新的些微改進，包括：

- 全新的 I/O 系統（在第四章〈基本宣告及運算式〉中會說明基礎知識。新的檔案系統在第十六章〈檔案輸入／輸出〉中詳細討論）。
- 常數及參考變數（在第五章〈陣列、修飾子以及讀取數值〉中說明）。
- 函式重載、inline 函式、參考參數及預設參數（請參考第九章〈變數範圍與函式〉）。

因此你可以將 C++ 當作更好的 C 來使用。但是 C++ 還增加了一些全新的特性，比如物件、模板和異常。所以從第十三章 " 簡單類別 " 開始，就會學習全新的概念。

字型體裁

本書使用以下的字型體裁：

斜體字（Italic）

用於目錄與檔名。範例中註解需強調的部分也會以斜體字表示。

粗體字（Bold）

使用在表示 C++ 關鍵字，以及第一次提及的新名詞和概念。

定寬字（constant width）

使用在表示程式和程式的元素，以及範例中顯示檔案的內容或命令的輸出。在正文中引用範例或程式片段的單詞或項目也以定寬字表示。

定寬粗體字（constant bold）

在範例中用來顯示命令或者其他需要由使用者逐字輸入的文字（例如，**rm foo** 表示當它出現在內文或範例中時，要鍵入 "rm foo"）。

定寬斜體字（Constant italic）

用在例子中表示應視情況取代的變數（例如，變數 **filename** 應該用某個實際的檔名來代替）。

" 雙引號 "

用以標示程式說明段落中的系統訊息或程式片段。

% UNIX 中 C shell 的提示符號。

$ UNIX 中 Bourne shell 或是 Korn shell 的提示符號。

[] 程式語法說明中的選項（括號本身不必輸入）。

… 代表為了使空間看起來簡明而省略的文字（通常是用來指電腦的輸出）。

CTRL-X 或是 ^X 指出使用控制字元，它代表按住 "Control" 鍵後再按下 "x" 鍵。我們以類似方法表示其他鍵（例如：RETURN 代表換行）。

若沒有特別指出，則在下命令後應按下 RETURN。

批評和建議

雖然我們已盡最大可能測試驗證過本書中的資訊，但你可能發現到書中所描述某種功能實際上已經改變了，或甚至根本是我們搞錯了。請將你所發現的錯誤，還有你對本書未來版本的任何寶貴意見告訴我們。來信請寄：

美商歐萊禮股份有限公司台灣分公司

電話：(02)2709-9669

傳真：(02)2703-8802

網址：*http://www.oreilly.com.tw*

電子郵件：

sales@oreilly.com.tw（業務部）

editors@oreilly.com.tw（編輯部）

bookquestion@oreilly.com.tw（疑難雜症）

請以電子郵件的方式與我們聯絡，這會比電話和舊有的郵件方便。如果你買到的書有印刷品質上的問題，可以寫信到業務部；若你對書籍內容有疑義，或是發現錯字，請寫信到 *bookquestion@oreilly.com.tw*。

O'Reilly 的每一本書都有專屬網頁，你可以在此找到關於本書籍的相關資訊，包括範例程式的下載、勘誤表與相關資源的連結。

http://www.oreilly.com/catalog/cplus2/　（本書英文版的網頁）

http://www.oreilly.com.tw/chinese/c/cplus2.html　（本書中文版的網頁）

第一版致謝

感謝 Peg Kovar 幫助我校對和編輯。特別感謝 Dale Dougherty 將我的第一本書弄亂後，逼我再將它正確地組裝在一起。非常感謝 Phil Straite 和 Gregory Satir 的辛勤付出。尤其感謝所有審查和編輯本書的人。還感謝 O'Reilly & Associates 公司製作？品組的專案經理和編輯 Nicole Gipson，製作助理 John Files、Juliette Muellner 和 Jane Ellin，以及書籍設計執行人 Mike Sierra。最後，特別感謝那些辛苦的程式設計師，他們編寫的程式碼使我受益匪淺。

第二版致謝

對於第二版，我希望感謝我的編輯 Robert J. Denn，他為本書的出版付出了耐心和艱苦的工作。感謝 Ray Lischner 的技術指導。Al Stevens 因其廣泛的 C++ 知識和嚴格的標準，應受到特別的讚賞。他的努力有助於我強化了術語及細化了書中的範例，從而使得本書的原稿更加精確。本書的任何錯誤都是我自己造成的，而不是審稿者或 O'Reilly 的工作人員的責任。

我還要感謝 O'Reilly 的所有銷售和市場開發人員，他們拼命地工作來銷售我的書。

關於中文版

本書中文版第二版係以第一版（呂維毅譯）為基礎，進行編譯更新。