
序

這本書是專為實作 C++ 程式設計而寫，不只是會告訴您這種語言的運用技巧，也會說明寫作風格與除錯方法。內容討論到程式發展的整個循環，包含了觀念、設定、除錯、上線、文件、維護及修改等各角度。

風格是本書特別強調的重點。寫一個好的程式不應只是程式而已，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。優良的程式不只是運作正常，而應簡單容易被人了解。註解是讓程式設計人員將說明的文字放在程式中來解說，一個說明清楚、明白的程式十分重要。

程式應儘可能的簡單，避免用別人不易瞭解的小技巧。不易瞭解的小技巧與複雜性會使程式變得一無是處。這本書強調的是簡單、實用的法則。例如，在 C++ 之中 15 個運算子的優先順序其實可以簡化為二點即可說明完畢：

1. 先乘除後加減。
2. 將括號放在其他各個位置。

試想有二個程式，其中一個程式是絕頂聰明的程式設計師所寫的，？頭有很多別人不易了解的小技巧。這個程式沒有任何註解說明，但是可以運作。另一個程式的註解很清楚並且是結構化的組織，但是有些小問題。那一個程式會比較有用？就長期的角度來看，有問題的程式較佳，因為它可加以進行修改並且維護容易。雖然用小技巧的程式目前運作正常，但是遲早會需要加以修改。你會發現遇到的最困難工作會是修改不易了解的程式。

本書的內容

這本書的內容適合沒有任何程式設計經驗的人、只了解 C 語言但想要了解 C++ 的程式設計師，以及已知道 C++ 但要提昇程式設計風格與可靠度的人。你必須有電腦來進行實際操作，以及知道基本的使用，像是編輯程式 (editor) 及檔案的運作處理。

學習電腦語言最快的方式是寫程式與除錯。在凌晨花二個小時的時間去找出有問題的程式對你很有幫助，即使是發現到輸入的 '=' 符號應該是 '==' 符號才對。在這本書中會有許多程式設計的範例加以說明，有的還會故意放入錯誤。你應把程式輸入到電腦，並加以執行與進行除錯。這個程序會讓你在小程式中發現常見的錯誤，因此你會學到如何在自己的程式中去找出及修改這些錯誤 (如何取得本書中的程式會在下面加以說明)。

有多個專業的 C++ 會在後面加以說明：

- 一般的 UNIX 編譯程式，可在大多數的 UNIX 系統上運作。
- GNU C++ 編譯程式，名稱是 g++ (在大多數的 UNIX 系統上都可見到)。
- Borland 公司在 MS-DOS 上的 Turbo C++ 編譯程式。
- 在 MS-DOS/Windows 的 Borland C++。
- 在 MS-DOS/Windows 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程式只有些許的不同。這本書會清楚說明編譯程式的不同與應注意的地方，在用各個編譯程式時會特別解說關於如何在這些編譯系統上加以編寫與執行。這本書之中也會有介紹如何使用程式設計工具 make 來自動進行處理。

本書的組織方式

在學會走之前應先學會爬。在第一部份：“基礎”中，你將學會如何爬行。這些章節會教你基本程式的設計方法，你會從程式設計的功能與設計風格開始。接著，會學到如何使用變數與簡單的判斷敘述和控制命令。

此時你已學會如何建立最基本的程式；在第七章“程式開發的過程”中，你會了解到整個設計的程序，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“程式寫作的基礎”。

說明基本程式設計的程序，以及如何寫一個簡單的程式。

第三章“風格的展現”。

討論程式設計的風格。如何說明程式以及如何寫清楚易懂的程式碼。

第四章“基本宣告及運算式”。

說明簡單的 C++ 敘述，包含基本變數及指定命令的細節，以及算術運算子：
+、-、*、/ 與 %。

第五章“陣列、修飾子以及讀取數值”。

包含陣列及其他較複雜的變數形態，及縮寫運算子如 ++、--、*=、=、+=、-= 與 %=。

第六章“條件式及控制敘述”。

說明簡單的判斷敘述，含有 if、else 及 for。另外會討論 '=' 與 '==' 二者的差異。

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，從規格的制定到完成。討論的內容有結構化程式設計、快速原形設計及除錯。

第二部份“程式設計基礎”，

說明其他程式設計的簡單敘述及運算子。你也會學到如何組織這些敘述成為簡單函式。

第八章“進一步的控制敘述”。

說明其他的控制敘述，包含 while、break 及 continue，也會討論 switch 敘述的細節部份。

第九章“變數範圍與函式”。

說明區域變數、函式及參數。

第十章“C++ 前置處理器”。

說明 C++ 前置處理器，可在設計程式碼時具有很大的彈性，但是它也會很容易讓你犯錯。有些簡單的規則可使前置處理器避免發生錯誤。

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。

第三部份“進階的形態及類別”，

你將會學到建立高等形態如結構、union 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“進階的形態”。

會說明結構及其他高等形態，像是 sizeof 運算子及 enum 形態。

第十三章“簡單的類別”。

說明類別的基本概念，這是 C++ 最強的特性。類別允許你將資料與運算結合，將資料結合成一個物件。

第十四章“再論類別”。

說明可用類別處理的其他運作。

第十五章 “指標的基礎”。

說明 C++ 指標變數與解說使用方法。

第四部份 “進階程式設計的概念”。

談論高階的程式設計技術。在這個部份將介紹多個 C++ 特性，可讓你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”。

說明緩衝處理及未緩衝處理的 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。傳統 C 方式的 I/O 處理也會說明。

第十七章 “除錯及最佳化”。

會解說如何為程式除錯，以及如何使用互動式除錯程式。不但會說明如何對程式除錯，同時也指出如何寫一個容易除錯的程式。這一章內容還會說明許多最佳化的技術，使程式執行更快及更有效率。

第十八章 “運算子重載”。

C++ 允許你定義語言的運算子來擴充該語言的處理，在這章中，你可學到如何建立一個複雜形態與處理複雜形態的運算子。

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所會發生的問題，像是計算的誤差、精確度的問題、overflow 及 underflow。

第二十章 “進階指標”。

介紹以指標的高階使用來建立動態結構，例如：連結串列及樹。

第二十一章 “進階的類別”。

說明如何從簡單的基底類別來建立複雜、衍生類別。

第五部份“其他的語言特性”。

說明其他各項特性。

第二十二章“例外”。

說明如何在程式中處理預期之外的情形。

第二十三章“模組化程式設計”。

敘述如何將程式分成多個檔案及使用模組化程式設計技術，並介紹公用程式 make 的細節。

第二十四章“模版”。

允許你定義通用函式或是類別，來產生一系列的函式。

第二十五章“移植性的問題”。

說明移植程式時（從一個機器搬到另一個機器上執行）可能發生的問題。

第二十六章“整合”。

敘述一個複雜程式從開始有一個想法到完成之間所需要的每個步驟細節，特別強調資訊處理和模組化程式設計技術，以及物件導向程式語言。

第二十七章“從 C 到 C++”。

說明如何將 C 程式轉成 C++ 程式，以及說明 C 程式會影響到 C++ 程式設計人員的陷阱。

第二十八章“C++ 的被遺忘角落”。

說明 do/while 敘述、逗號運算子及?: 運算子。

第二十九章“程式設計的規則”。

列出可幫助你寫出好程式的程式設計規則。

第六部份 “附錄”

附錄 A “ASCII 表格”。

包含有一連串字元碼及其值。

附錄 B “變數範圍”。

列出部份 C++ 變數形態的數值範圍。

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法則。

附錄 D “使用乘方級數來計算 sine”。

說明電腦如何處理 sine 函式的計算程式。

如何讀這本書 (若你已學會 C 語言)

C++ 是以 C 語言為基礎，如果你已經學會 C 語言，你將會發現到第二章到第十二章的內容只是在說明 C 語言。

C++ 含有許多新的特點，像是：

- 全新的 I/O 系統。（在第四章“基本宣告及運算式”中會說明基本的觀念。新的檔案系統會在第十六章“檔案輸入/輸出”中詳細說明。）
- 常數及參考變數。（請看第五章“陣列、修飾子以及讀取數字”。）
- 函式重載、inline 函式、參考參數及預設參數（參考第九章“變數的範圍及函式”）。

使用的字型解說

括號中的字體用來指出系統訊息或是說明的程式。

%

這是 UNIX 中 C shell 的提示符號。

\$

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

#

這是 UNIX superuser 的提示符號 (不論是 Bourne shell 或是 C shell), 我們會使用這個項目限定由 root 才能執行時。

[]

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 (例如: ENTER 代表換行)。

若沒有特別指出, 則在下命令後應按下 ENTER。

如何取得原始程式

你可從 Internet 上向 O'Reilly & Associates 取得這本書中的原始程式碼。

本書中的範例可用多種方式取得，像是 FTP、Ftpmail、BITFTP 及 UUCP。這？依最便宜、最快速及最容易的方法來排列其說明順序。如果你依序去看，那麼第一個可使用的就是最佳的。如果連接在 Internet 上，可使用 FTP 的方式。若沒有直接在 Internet 上，則可用 Ftpmail 來在 Internet 進行送與收電子郵件。或是也可在 BITNET 上用 BITFTP 來傳送電子郵件。若是前幾種方式都不行的話，則可使用 UUCP。

FTP

採用 FTP 的方式必須要有部電腦接上 Internet。下面是個說明的範例，你應輸入的命令會用粗體字表示。

```
% ftp ftp.uu.net
Connected to ftp.uu.net.
220 FTP server (Version 6.21 Tue Mar 10 22:09:55 EST 1992) ready.
Name (ftp.uu.net:joe): anonymous
331 Guest login ok, send domain style e-mail address as password.
Password: joe@ora.com (請在這裏輸入你自己的使用者名稱及主機名稱)
230 Guest login ok, access restrictions apply.
Ftp> cd /published/oreilly/nutshell/practcpp
250 CWD command successful.
Ftp> binary (這很重要！壓縮檔案必須指定為二進位模式的方式傳送)
200 Type set to I.
Ftp> get examples.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for examples.tar.gz.
226 Transfer complete.
ftp> quit
221 Goodbye
%
```

這個檔案是用經過壓縮的 tar 形式；要取出檔案可輸入下面的指令：

```
% gzcat examples.tar.gz | tar xvf -
```

System V 的系統要用下面的方式：

```
% gzcat examples.tar.gz | tar xof -
```

若是 gzcat 在你的系統中無法使用，可使用分開的 gunzip 及 tar 或 shar 命令。

```
% gunzip examples.tar.gz  
% tar xvf examples.tar.gz
```

Ftpmail

Ftpmail 是一個電子郵件的伺服器，可在 Internet 上送與收電子郵件，可讓任何公司或是服務提供者傳送電子郵件到 Internet。

你可傳送郵件到 ftpmail@online.ora.com，在訊息的部份放入要執行的FTP命令。伺服器會執行不具名（anonymous）的 FTP 傳送，並將取得的檔案用郵件處理的方式傳回。要取得完整的輔助說明，可傳送一個空主題的訊息，並在內文放入“help”這個字。

下面是一個簡單的範例。這個命令會傳送指定目錄中的檔案給你。

```
% mail ftpmail@online.ora.com  
Subject:  
reply-to janetv@xyz.com (你要求檔案傳送的電子郵箱)  
open  
cd /published/oreilly/nutshell/practcpp  
mode binary  
uuencode  
get examples.tar.gz  
quit  
.
```

在訊息的最後也可加上你的簽名資料，但必須是在“quit”之後。

BITFTP

BITFTP 是供 BITNET 使用者所使用的郵件伺服器。你可用來傳送電子郵件的訊息與取得檔案，並且它會用電子郵件的方式傳回。BITFTP 目前只供 BITNET/EARN 與 NetNorth 連接的使用者。BITFTP 是 Princeton University 所提供的公用服務，下面是它處理的方式。

使用 BITFTP 的方式，可將含有 ftp 命令的郵件傳送到 BITFTP@PUCC。若要取得完整的輔助說明檔，可在訊息內容中加入 HELP 這個字。

下面是要傳送到 BITFTP 的訊息內容：

```
FTP ftp.uu.net NETDATA
USER anonymous
PASS myname@podunk.edu (用你的Internet電子郵箱而不是BITNET的位址)
CD /published/oreilly/nutshell/practcpp
DIR
BINARY
GET examples.tar.gz
QUIT
```

在取得指定的檔案後，依 FTP 的指示來取出檔案。因為你可能不是在 UNIX 系統，所以需要系統中提供有 uudecode、uncompress、atob 及 tar 命令。VMS、DOS 及 MAC 都可以找到這些處理的程式。

UUCP

UUCP 幾乎在每個 UNIX 系統中都會有，並且 IBM 相容的 PC 及 Apple Macintoshes 都可找到。經由數據機從 UUNET 上以 UUCP 的方式取得；UUNET 的連接會需要付費。

不論你在 UUNET 上是否有帳戶都可以取得範例。如果你或是你的公司在 UUNET 上有帳戶，可以直接用 UUCP 連接到 UUNET。請輸入：

```
uucp uUNET\!~/published/oreilly/nutshell/practcpp/examples.tar.gz
yourhost\!~/yourname/
```

若是使用 sh (Bourne shell) 而不是 csh , 則可以省略反斜線。這個檔案經過一段時間後會出現在目錄 /usr/spool/uucppublic/youname。如果你沒有帳戶, 但是希望要申請一個使用電子郵件的帳戶, 可與 UUNET 連絡, 電話是 703-204-800 (美國)。

/published/oreilly/s-IR.Z 這個檔案中含有全部檔案的大小與名稱。

在找到需要的檔案後, 依照 FTP 上的指示傳回需要的檔案。

你的建議

關於本書內容的建議和問題, 你可與出版公司連絡:

O'Reilly & Associates, Inc.

103 Morris Street, Suite A

Sebastopol, CA 95472

1-800-998-9938 (在美國或是加拿大地區)

1-707-829-0515 (國際)

1-707-829-0104 (FAX)

誌謝

感謝 Peg Kovar 幫忙編輯校對這本書; 特別要感謝 Dale Dougherty 逼我將這本書的內容正確的組合在一起; 很感謝 Phil Straite 為了這本書所付出的辛勤努力; 尤其要謝謝所有曾經閱讀、批評、編輯這本書的人。除此之外, 還要謝謝 O'Reilly & Associates 的專案經理 Nicole Gipson, 催生這本書的 John Files、Juliette Mueller、Jane Ellin, 和設計這本書外觀的 Mike Sierra。最後, 感謝所有正在奉獻自己青春的程式設計師們, 他們的程式碼令我受益匪淺。