

譯序（侯捷）

做為一項重要的軟體技術，OO（Object Oriented）已經盛行多年，日益成熟。

萬丈高樓平地起，OO 的一大堆技術、思維、分析、設計…，最終要落實到編程動作（programming）來。OOP（Object Oriented Programming）成了踏入 OO 領域的最根本。

全世界的 OO 程式語言中，目前以 C++ 的接受度最廣，使用人口最多，近年來甚至被國內各大專資訊科系視為必修課程。一本好的 C++ 教科書，對於軟體尖兵的技術培養有很不小的貢獻。

不論做為上課或自修，C++ *Primer* 都是一本很好的 C++ 教科書。雖然它「是否適合初學者」的正反意見始終不斷，但這類討論的癥結在於每個人所謂的初學者意義都不相同，這類討論其實沒有交點，也沒有意義。

我於 1999 年將 C++ *Primer* 翻譯為《C++ *Primer* 中文版》。其中有許多突破性的翻譯理念和製作理念（詳見該書導讀，或 <http://www.jjhou.com/cpp-primer-chap0.pdf>），獲得許多好評。此書讀者大致可分兩類，第一類把它當做工具書（查閱用），第二類把它當做學習書。前一類讀者當然不需要書中的練習題，因為他們都已能熟用 C++ 語言。後一類讀者可能很需要實作書中的練習題，以便驗證自己的學習並提昇自己的功力。

Tondo & Leung 所著的這本 C++ *Primer Answer Book*，獲得了 C++ *Primer* 作者 Lippman & Lajoie 的認同，可謂是一本官方解答。為了讓上述第二類讀者在練習時有一個憑藉，我決定將此一解答本也中文化。

本書由我和黃向陽先生合作。向陽是我的學弟，也是我的學生，他曾不辭勞遠地每週從新竹到內壢旁聽我的課程，向學精神與積極態度令我印象深刻。向陽目前是大四學生，以學生的身份以及優於一般學生的身手進行這一本對許多學生而言有重要影響的「習題解答」翻譯工作，我想至為適合。一般所

謂合譯是「你譯一半，我譯一半，兩不相干」，這是最詬病的一種方式。本書由向陽完成全部初稿，再由我完成全部的技術檢閱與文字修潤。我們都希望將書籍做到完美，但是人世間沒有完美。請上本書網站（載於封底）觀看後續的討論、勘誤、程式範例。

侯捷 2000.10.05 于新竹

jjhou@ccca.nctu.edu.tw

<http://www.jjhou.com>

<http://jjhou.readme.com.tw>

p.s. 本書對照原文本，採頁頁對譯方式，俾得以保留原文索引。

溫故知新（黃向陽）

兩年前，我們藉《多型與虛擬》掌握了 C++ 的核心觀念，一年前，我們藉《C++ Primer 中文版》窺得 C++ 的全貌，徹底打破了語言的隔閡。現在，我們藉由這本《C++ Primer 習題解答》，再度踏上那曾經走過卻可能不再熟悉的道路。

C++ 語言的創造者 Bjarne Stroustrup 在接受 C++ Report 總編輯 Herb Sutter 的獨家專訪時曾說：『我認為閱讀和撰寫程式碼，對於專業程式員的訓練，極為重要。』*The C++ Programming Language* 和 *C++ Primer* 這兩本著名的 C++ 百科書籍都精心設計了練習題，也就因此毫不令人意外了。研讀完一小段語法之後，動手寫個小程序，對於熟悉語法頗有助益。習題的作用在於澄清觀念、避免易犯的錯誤；雖不必太嚴肅看待，卻也絕對輕忽不得。

翻譯本書的過程中，我又把《C++ Primer 中文版》從頭到尾徹底地詳讀了一遍。作為本書譯者，我盡量使本書風格與《C++ Primer 中文版》一致，對於各個中文術語，都以《C++ Primer 中文版》的中英術語對照表為本。我從不認為大量中英文夾雜所造成的「破碎」版面，會是一件令人不舒服的事。反而我害怕閱讀一本全是中文字的翻譯書。在侯捷網站上，關於《C++ Primer 中文版》的讀者回應，有兩點我深感贊同：

1. 翻譯的難處在於長句子的處理。
2. 不必為英文術語硬取一個中文 alias（別名），因為就算取得再有聯想性，讀者還是會回頭去想原文究竟是什麼。

電腦翻譯書的目的在於掃除閱讀上的障礙，使讀者專注於技術上的思考。

工欲善其事，必先利其器。在著手進行一個 C++ 程式之前，請先準備好您的編譯器並將環境設定妥當。以下網址可免費下載 Win32 C++ 編譯器：

<http://www.borland.com/bcppbuilder/freecompiler>

環境設定方面，可參考侯老師的【在 console mode 中使用 C/C++ 編譯器】一文：

<http://www.jjhou.com/article99-10.htm>

感謝侯老師給我機會接觸書籍翻譯工作。從讀者轉變為譯者，我的心境有很大不同。書籍製作的背後實在有太多的瑣事，每一項都令人費神。我認同電腦技術作家必須具備排版能力，因為我見過一些很好的作品，經由所謂「排版人員」的處理後令人不敢卒睹。但我也並不否認，在一個分工的社會，如果能省下這些繁瑣的工作，必定能讓作家們更有生產力，將心思專注於技術的鑽研與創作。

我曾和一位電腦技術作家有過一次交談機會，他提出技術排版的觀念。所謂技術排版人員，理想上必須對書中內容有概略的認識，知道哪些是術語、哪些是專有名詞。以電腦書籍而言，尤其必須知道**程式碼該排等距字！**

如果這樣的觀念能獲得重視，我想國內電腦書籍的品質會更上一層。

個人的期許

C++ 提供了多種不同的程式設計思維模式，其中以 object-oriented 和 generic paradigm 最為重要。兩大技術皆與軟體的重複可用性密不可分。它們分別運用了多型與泛型；多型仰賴 virtual function 實現，泛型憑藉 template 完成。

在純粹探究 C++ object-oriented 程式設計方面，我們已經有了一本中文好書。Generic paradigm 呢？我想，我們期待著傳聞中的那一本。

黃向陽 2000.09.30 于風城交大

syhuang.cis86@D2.nctu.edu.tw