電腦、網際網路與全球資訊網簡介

1

學習目標

在本章中，你將學到：

- 電腦基本概念
- 不同類型的程式語言
- C 程式語言的演進歷史
- C 標準函式庫的功能
- 物件技術的基礎
- 典型的 C 程式開發環境元件
- 在 Windows、Linux、OS X 的 C 語言試行應用
- 網際網路與全球資訊網沿革

本章綱要

摘要｜自我測驗｜自我測驗解答｜習題｜進階習題

1.1 簡介

歡迎來到 C 與 C++！C 是一種簡潔但功能強大的程式語言，適合稍微有程式設計經驗或是完全沒有程式設計經驗的技術人員，也適合資深的程式設計師建立大量的軟體系統。對這些人來說，《C 程式設計藝術》(第七版)是極佳的學習工具。

本書的核心在於透過已實證之 C 的結構化程式設計與 C++的物件導向程式設計方法，來強調軟體工程。本書提供數以百計的可執行程式，並且列出這些程式在電腦上執行的結果。我們稱之爲「實況程式碼」(live-code)教學。所有的範例程式都可以從我們的網站 **www.deitel.com/books/chtp8/**上下載。

大部分的人都知道電腦能執行許多有趣的工作。使用這本書，你將學會如何命令電腦來執行這些工作。電腦（通常稱作**硬體**，hardware）由**軟體**（software）所控制（軟體就是人撰寫的指令，可讓電腦執行**動作** [action]並做**判斷** [decision]）。

1.2　硬體與軟體

電腦可用來執行計算和做出比人類更快的邏輯判斷。現今許多個人電腦可以在一秒內執行十億個計算，遠超過人類一輩子所做的運算。超級電腦已經可每秒執行數以千兆（10 的 15 次方）個指令！中國國防科技大學的天河二號超級計算機每秒可以執行超過 33 萬億次計算（33.86×10^{15}）！[1]從這個角度來看，天河二號超級計算機可以在一秒鐘內爲地球上每個人完成約 300 萬次計算！超級計算的「上限值」正在快速增加中！

電腦受到稱爲**電腦程式**（computer programs）的一組指令控制來處理**資料**（data）。這些電腦程式指揮電腦去執行一連串有次序的動作，這些動作都是由**電腦程式設計者**（computer programmers）預先指定的。

電腦是由各種裝置（如鍵盤、螢幕、滑鼠、硬碟、記憶體、DVD 以及處理器）所組成的，我們稱它們爲硬體（hardware）。由於軟硬體的技術快速發展，電腦的價格急速地下降。在幾十年前，體積龐大到塞滿整個大房間，費用高達數百萬美金的電腦，現在竟然能夠放在一個比指甲還小的矽晶片表面上，單價只要幾塊美金。幸運的是，矽是地球上藏量最豐富的物質之一，也是構成沙子的主要元素。矽晶片技術造就了運算處理是非常便宜的，以至於電腦已經變成了商品化。

1.2.1　摩爾定律（Moore's Law）

每年，人們通常想以多一點點的錢來得到更多的產品及服務。相對的情況表現在電腦和通信領域，特別是關於硬體以及支援這些技術的硬體費用。數十年來，硬體的規格一再地快速滑落。每隔一兩年，電腦運算能力會增加近一倍，且仍廉價。這個出色的趨勢通常稱做**摩爾定律**，以提出此定律的人——Intel 的共同創辦人 Gordon Moore 來命名。Intel 是現今電腦與嵌入式系統內處理器的領導者。摩爾定律和類似的趨勢在某些事物上特別地明顯：像是電腦的記憶體數量（用來提供程式所需）、輔助儲存裝置的容量（如磁碟機，可用來長時間儲存資料）、以及處理器速度（與電腦執行程式速度有關）。同樣的成長現象也發生在通訊領域，硬體的價格直線下降，特別是最近幾年通訊頻寬的大量需求與競爭之下。沒有其他領域的技術進展地如此迅速。如此驚人的發展眞眞實實地促進了資訊革命。

---

1. http://www.top500.org.

1.2.2 電腦的架構

不管電腦外觀爲何，每部電腦實際上都可分成不同**邏輯單元**（logical units）或區段（圖 1.1）。

邏輯單元	描述
輸入單元 (input unit)	這個用來「接收資訊」的區域，會從**輸入裝置**（input devices）取得資訊（資料與電腦程式），並將此資訊放在其它裝置上以進行處理。大多數的訊息是透過鍵盤、觸控螢幕與滑鼠來輸入到電腦中。其他的輸入型式包括有接收語音指令、掃描影像與條碼、從次儲存裝置（如硬碟、DVD 磁碟機、藍光光碟機與 USB 快閃儲存記憶體——也可稱做「大姆哥」或「隨身碟」）、從網路攝影機接收影像和透過你的個人電腦上網接收資料（如你從 YouTube®下載影片檔或從 Amazon 下載電子書）。更新的輸入型態包含 GPS 來的地理位置、從智慧型手機與遊戲把手中的加速計（一個可以回應上/下、左/右、前進/後退加速的設備）提供的動作與方向訊息（例如 Xbox®的 Microsoft® Kinect®、Wii Remote™ 與 Sony® PlayStation® Move）。
輸出單元 (output unit)	這個用來「輸出資訊」的區域會接受經電腦處理過的資訊，並且將資訊送到不同的**輸出裝置**（output devices），讓這些資訊能夠在電腦之外使用。現今大多數的電腦會將資訊輸出到螢幕上（包括觸控螢幕）、列印到紙張（「綠色環保」並不鼓勵此行爲）、在個人電腦上播放聲音或視頻（例如 Apple 有名的 iPods）、體育館的超大型螢幕、在網路上發布或使用來控制其他裝置，如機器人與「智慧型」家電。資訊也通常輸出到輔助儲存設備，例如硬碟、DVD 驅動器和 USB 快閃儲存裝置。最近流行的輸出形式是智慧型手機和遊戲控制器振動，以及 Oculus Rift 之類的虛擬實境設備。
記憶單元 (memory unit)	這是電腦中一個存取快速、但容量相對較低的「倉庫」區域。它將輸入單元接收到的資訊保存起來，需要時，可以馬上運用這些資訊。記憶體單元也將電腦處理過的資訊儲存起來，直到該資訊可透過輸出單元送到輸出裝置爲止。記憶體中的資訊是「揮發的」（volatile）——當電腦的電源關掉時，裡面的資訊就會消失了。記憶體單元又稱爲**記憶體**（memory）、**主記憶體**（primary memory）或 RAM（**隨機存取記憶體**，Random Access memory）。一般桌上型或是膝上型電腦的主記憶體內建 128GB 的 RAM，2GB 到 16GB 是最常見的。GB 是 GigaByte 的縮寫，1GB 接近十億個位元組。一個**位元組**（bytes）是 8 位元。位元不是 0 就是 1。

圖 1.1 電腦的邏輯單元(1/2)

字元（characters）

對於人們來說使用低階的位元來處理資料是很乏味的，取而代之，偏好使用十進位數字（0 至 9）、字母（A–Z 和 a–z）、特殊符號（例如：$、@、%、&、*、(、)、–、+、"、:、?和/）。數字、字母、和特殊符號是所謂的**字元**，電腦的**字元集**是用來撰寫程式與代表資料，電腦僅處理 1 與 0，因此所有字元都可以用 1 與 0 的格式來表達。C 語言支援多種字元集（包括**萬國碼字集** [Unicode®]），由一個、兩個或四個位元組組成（8、16 或 32 位元）。萬國字碼集包含世界上許多語言的字元。更多關於 ASCII **字元集**的資訊請參考附錄 B——萬國碼的常用子集合，包括大小寫字母、數字和一些特殊字元。

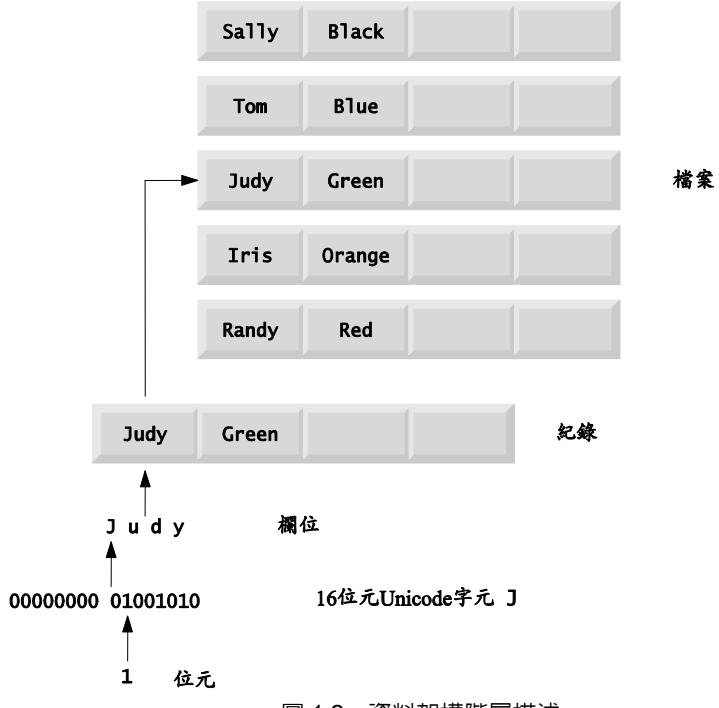

圖 1.2 資料架構階層描述

欄位（fields）

如同字元是由位元所組成，**欄位**是由字元或位元組所組成。欄位是一組有意義的字元或位元組的集合，例如，一個欄位包含了大小寫字母用來表示人名，並且另一個欄位包含著數字來表示人的年齡。

紀錄（records）

記錄是以數個欄位所組成。在一個薪資系統裡，舉例來說，對於一個員工的紀錄可能包含著下列欄位（括弧中表示欄位可能的資料類型）：

新的 C 語言標準

在此也介紹新的 C 語言標準（或稱爲 C11），2011 年時公布。新的標準加強並延伸 C 語言的能力。本書將這些建構了新功能的編譯器包含在附錄 E（可參考或省略的章節）。

可攜性的小技巧 1.1

由於 C 是與硬體無關且被廣泛使用的程式語言，所以用 C 寫的應用程式只需稍加修改，甚至不需修改，就可在許多不同的電腦系統上執行。

1.6　C 標準函式庫

你將會在第五章中學到，C 程式是由許多稱爲**函式（functions）**的模組或是片段所構成的。你可以自己設計所有要用到的函式，不過大多數的 C 程式設計師都會利用一套現成的函式，稱爲 **C 標準函式庫**。所以，學習 C 語言可分爲兩個部分，第一是學習 C 語言本身，第二則要學習如何使用 C 標準函式庫所提供的函式。透過本書我們將探討這些函式。P.J.Plauger 的書《C 標準函式庫（The Standard C Library）》對於想要更深入瞭解函式庫、如何建構、如何運用寫出可攜性程式碼的的程式設計師們是必讀的。我們使用到許多這本書中介紹的 C 函式庫函式。

《C 程式設計藝術（第八版）》倡導以區塊導向編寫程式。避免重寫軟體。因此使用現有的區塊則稱作爲**軟體再用**。當使用 C 語言撰寫程式，將會使用到下列構件：

- C 標準函式庫所提供的函式。
- 自己撰寫的函式。
- 其他人撰寫並提供給你使用的函式。

自行建立函式的優點，是可以清楚瞭解其實際運作方式。你可自行試驗 C++原始碼。缺點則是必須花費時間來設計、發展、偵錯新函式，並對新函式進行效能調整。

增進效能的小技巧 1.1

盡量利用 C 標準函式庫所提供的函式，而不要自己撰寫。因爲標準函式都是經過專家精心設計的，效率會比較好。

可攜性的小技巧 1.2

盡量利用 C 標準函式庫所提供的函式，而不要自己撰寫。因爲標準函式適用於所有標準 C 的實作，所以可攜性較高。

1.9.5 第六階段：執行

最後，電腦會在 CPU 的控制下，以每次執行一個指令的方式開始**執行**（executes）程式。在 Linux 系統中，載入和執行只需在 Linux 提示字元後鍵入**./a.out**，然後按下 Enter 鍵即可。

1.9.6 執行時可能會發生的問題

程式在第一次測試時不一定會成功。前面幾個階段都可能因各種錯誤而失敗，這些錯誤本書都會討論。例如：一個執行中程式可能會除以 0（如同算術，在電腦中這是一個非法的運算）。這可能會使電腦顯示一段錯誤訊息。若發生這樣的情形，就要回到編輯階段做些必要修正，再繼續後面的幾個階段，看看改對了沒。

常見的程式設計錯誤 1.1

當程式執行發生除以零的錯誤時，這種錯誤稱爲執行時期錯誤（run-time error 或 execution-time error）。除以零通常是致命錯誤，也就是使程式立即終止無法繼續執行的錯誤。非致命的執行時期錯誤（Nonfatal runtime errors）可以允許程式繼續執行完畢，但通常會產生錯誤的結果。

1.9.7 標準輸入、標準輸出和標準錯誤串流

C 中大部份程式都會輸入和/或輸出資料。大部分的 C 函式都從 **stdin（標準輸入串流，**standard input stream）輸入資料，**stdin** 通常是鍵盤，不過也可以重新指向到其他的裝置。大部分的 C 函式都從 **stdout（標準輸出串流，**standard output stream）輸出資料。**stdout** 通常是螢幕，不過他們也可以連結到其他的裝置。當我們說「程式印出結果」時，通常是指在螢幕上顯示結果。資料也可輸出到其它裝置，如磁碟和印表機。電腦也有**標準錯誤串流**（standard error stream），稱爲 **stderr**。**stderr** 串流（一般會連到螢幕）用來顯示錯誤訊息。使用者通常會將輸出資料 **stdout** 指定到螢幕以外的裝置，而 **stderr** 仍指定到螢幕，因此當錯誤發生時，使用者會馬上知道。

1.10 C 應用於 Windows、Linux 和 Mac OS X 的試行

本節之中，將會執行並與讀者的第一個 C 應用程式互動，這裡將會從執行猜數字遊戲開始，它會隨機地從 1 至 1000 抽出一個數字並請讀者猜猜看。若讀者猜對，遊戲即結束，若讀者猜錯了，應用程式會提示比正確數字高或是低。猜數字的次數沒有任何限制，但讀者應該是可以在 10 次或更少的次數猜出。在這個遊戲背後隱藏著一些有趣的電腦科學—章節 6.10，搜尋矩陣，讀者將會發現二元搜尋的方法。

自我測驗

1.1　填充題：

a)　電腦受一組稱作_____的指令控制來處理資料。

b)　電腦的重要邏輯單元是_____、_____、_____、_____、_____、_____。

c)　本章討論的三種語言爲、_____、_____。

d)　能夠將高階語言程式轉譯成機器語言的程式稱爲_____。

e)　_____是一種以 Linux 核心和 Java 爲基礎的行動裝置作業系統。

f)　_____軟體通常功能完整，原則上應該已沒有程式錯誤且可供社群使用。

g)　Wii 遙控器以及許多智慧型手機都使用了_____，讓設備能夠對運動做出回應。

h)　C 語言最初是以開發_____作業系統而聞名。

i)　_____是一種開發 iOS 和 Mac 應用程式的新型程式語言。

1.2　請在以下關於 C 環境的敘述中填空。

a)　C 程式通常是以_____軟體輸入電腦中。

b)　在 C 系統裡，_____程式會在進行轉譯過程開始之前自動執行。

c)　兩個最普遍的前置處理器命令是_____和_____。

d)　_____程式會將編譯好的程式碼和各種函式庫的函式結合，產生一個可執行影像檔。

e)　_____程式可以將 C 程式的可執行檔從磁碟傳到記憶體。

1.3　填充題（請見 1.8 節）：

a)　物件具有_____的性質—即使物件知道如何透過定義良好的介面來通訊，但通常不會讓他們知道其他物件是如何建構的。

b)　在物件導向程式語言，建立了一個稱爲_____的程式片段，來安排執行類別任務的方法。

c)　物件的新類別可被快速且方便地藉由_____來建立—新的類別包含來自於現有類別的特徵，可以將其客制化並增加獨一無二的特徵。

d)　尺寸、外型、顏色與重量都是屬於物件類別中的_____。

自我測驗解答

1.1　a) 程式　b) 輸入單元、輸出單元、記憶體單元、中央處理單元、算術邏輯單元、輔助儲存單元　c) 機器語言、組合語言、高階語言　d) 編譯器　e) Android　f) 最終測試版　g) 加速器　h) UNIX　i) Swift。

1.2　a)文書編輯　b)前置處理器　c)將其他檔案引入要編譯的檔案，將特殊符號以程式本文代換　d)連結器　e)載入器。

1.3　a) 資訊隱藏　b) 類別　c) 繼承　d)屬性。