單元 12

演算法與程式語言

單・元・目・標

- ☑ 瞭解演算法與流程圖的意義。
- ☑ 認識常見的程式語言。

12-1 演算法與流程圖

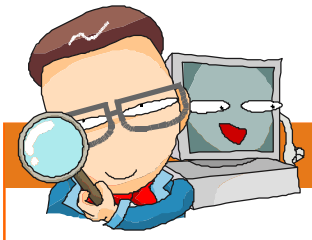
程式（program）是一連串邏輯性指令的集合。電腦之所以能發揮強大功能，替我們解決許多生活上的問題，乃是因為程式設計師針對各種問題想出適當的處理流程，並將之寫成一系列有順序的指令，電腦則依照這些指令的步驟逐一執行計算或是資料處理的動作，因而解決問題或產生我們所需的合理結果。而其中「一系列有順序的指令」指的便是「程式」。

12-1.1 演算法

如上所述，由於電腦只能依照「程式」指示逐步完成指定的工作，因此在設計程式時必須先將問題分解成許多小步驟，然後再依一定的次序逐步執行，而這個描述問題解決程序的方法便稱做演算法（algorithm），演算法必須使用正確的步驟才能達到解決問題的目的。舉例來說，下面是撰寫「計算十個整數之平均值」的演算法範例：

演算法的基本原則

1. 執行步驟是有次序的。
2. 每一步驟必須清楚描述。
3. 每一步驟必須確實可行
4. 能在有限的步驟內完成。



》》撰寫「計算十個整數之平均值」的演算法

想一想，「計算十個整數之平均值」應該如何做？

1. 求一列整數的平均值是以該列整數的和除以該列整數之個數。
2. 使用者必須逐一輸入10個整數。
3. 電腦逐一加總10個整數。
4. 將這10個整數的總和除以10，如此便可求得平均值。

所以將之寫成演算法後可得：

第一步： 程式開始。

第二步： 令 Sum＝0， Number＝0（Sum代表各整數的總和， Number則代表目前已經加總過的整數個數）。

第三步： 讀入一個整數。

第四步： 將這個整數的值加入Sum內（計算總和）。

第五步： 將 Number值加 1（因為多讀入一個整數）。

第六步： 判斷是否已讀入10個整數了？（Number是否等於10？）。

若已經讀入10個整數則跳到第七步。

否則跳到第三步（繼續處理下一個整數）。

接下頁

第七步： 計算平均值（亦即將 Sum 除以 10）。

第八步： 印出平均值。

第九步： 程式結束。

12-1.2 流程圖與虛擬碼

流程圖

雖然我們可以用演算法來說明解決問題的步驟，但是隨著問題的複雜性逐漸增加，光以演算法來說明解決問題的流程似乎不夠清楚，這時便可配合流程圖來增加演算法的說明性。所謂流程圖（flowchart）是指以各種特定的圖形符號來表示演算法，藉以說明處理方法與步驟的一種特定圖表，下面的表 12-1.1 則列出流程圖的一些常用符號。

表12-1.1
流程圖常用符號一覽表。

圖　形	名稱與說明
開始　結束	起迄符號 代表一個流程圖的開始與結束。
	處理符號 用一個矩形表示將執行的一些動作。
	決策符號 用一個菱形格子表示，代表著一種選擇或問題，必須以「是」或「否」來答覆它。
	輸入 / 輸出符號 用一個平行四邊形格子表示從外界輸入資料或把處理結果輸出。
	流向線符號 以直線配上箭頭表示各步驟被執行的先後次序，通常流向線的方向是由上而下，由左而右。

接下頁

圖　形	名稱與說明
	連接符號 以圓圈來表示連接流程圖的兩部份，例如將流向線轉移另一張紙時便可用連接符號表示流向線的連續。
	磁碟符號 用來表示儲存在磁碟中的資料。
	說明符號 在較複雜或需要說明部份，可以在該部份的左邊或右邊加上說明符號並且附上說明文字。

根據演算法直接撰寫程式並無不妥，但若能輔以圖形化的流程圖將更能減少錯誤發生並有效抓住重點，因此我們可以說「流程圖」是一種圖形語言，藉此可以更容易達到瞭解與溝通的目的。

舉個例子來說，我們若將之前「計算十個整數之平均值」的演算法繪製成流程圖，可得如圖 12-1.1 的結果。

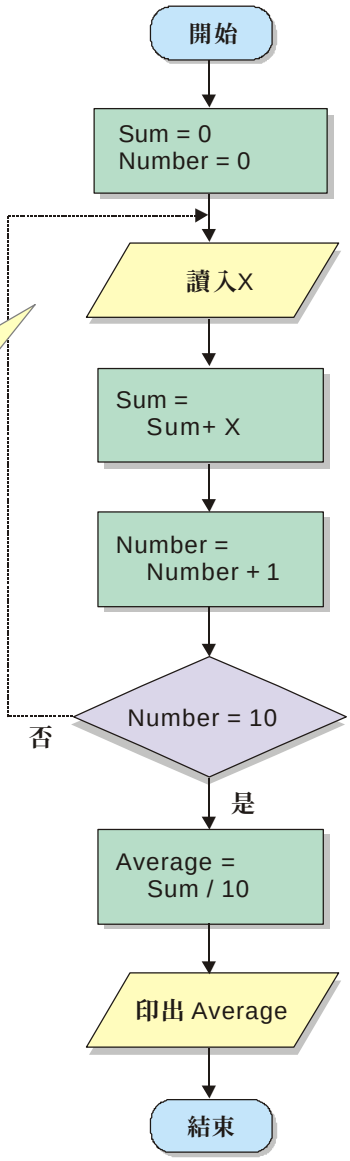


圖 12-1.1

「計算十個整數之平均值」的流程圖。

虛擬碼

虛擬碼（pseudo code）是一種介於一般語言與程式語言之間的語言碼。有些程式設計師不喜歡繪製流程圖，因此用一種很像程式碼的虛擬碼來表示程式的邏輯架構與執行程序，如圖 12-1.2，這些虛擬碼多半使用英文來表示，不過若使用中文也沒什麼不可以。虛擬碼並不能真的放到電腦中去執行，因為電腦看不懂這些虛擬碼。使用這些虛擬碼的最

大好處是，它很像在寫程式，這對於整天都在寫程式的程式設計師來說是最直接的一種思考方式。

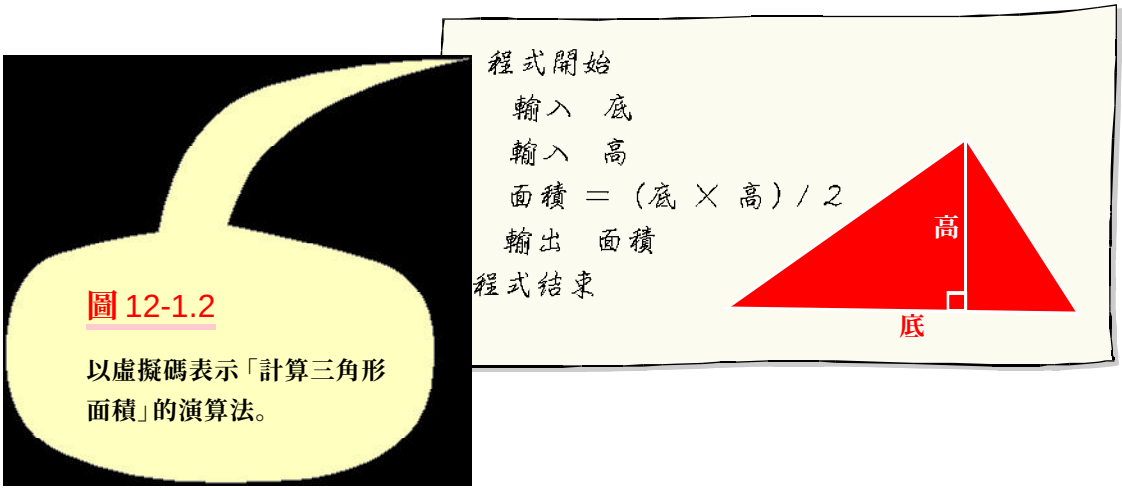
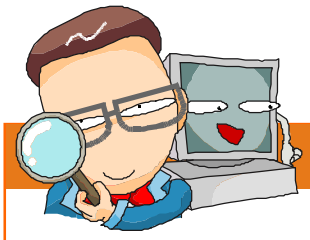


圖 12-1.2

以虛擬碼表示「計算三角形面積」的演算法。



》》撰寫「求三角形面積」的演算法並繪製流程圖

演算法：

1. 程式開始。
2. 輸入三角形的底和高。
3. 計算三角形的面積 = (底×高) ÷2。
4. 輸出計算後所得三角形的面積。
5. 程式結束。

流程圖繪製如右：