


認識電腦

電腦系統

商用電腦



張系國著

程式設計概論及基礎程式語言
計算機系統
資料結構及商用程式語言

認識電腦·電腦系統·商用電腦

作者：張系國

總策劃：林洋

總編輯：王麗

主編者：黃蘭

發行人：林大

出版：宏基出版社

總經銷：宏基書店有限公司

香港譚臣道一〇五號

印刷者：金源印刷廠

香港蘇杭街十九號

定價：港幣叁拾貳圓

有著作權·翻印必究

H.K.\$32.00

認識電腦 電腦系統 商用電腦

張系國著

序

一九六九年，我創辦了知識系統學院，一方面推動科際整合的研究工作，一方面訓練軟體工程學人材。到一九八一年夏季，就正式招生，我為學院的學生擬定了十門基本課程：

- CS1：程式設計概論及基礎程式語言（ BASIC ）
- CS2：計算機系統及科學工程用程式語言（ FORTRAN ）
- CS3：資料結構及商用程式語言（ COBOL ）
- CS4：計算機操作系統
- CS5：檔案結構及檔案管理
- CS6：資料庫管理系統
- CS7：層次程式設計及軟體設計原理
- CS8：軟體工程學原理

· 程式設計概論及基礎程式語言 ·

CS9：編譯程式技巧

CS10：軟體介面設計及軟體通訊原理

前三門課程，包括三種通用的程式語言 BASIC，FOR-TRAN 及 COBOL 的程式設計。雖然坊間程式設計的教科書汗牛充棟，我都不很滿意。一般的缺點，是太注意程式語言的細節，却不能清楚說明程式語言的基本通性，以及層次程式設計的技巧。我過去教的多半是研究院的課程，創辦知識系統學院後，不能不注意到程式語言的基本訓練。坊間課本既然都不合意，我乃決定編寫一套教材，供學院的學生使用。這套教材，除了在學院教授學生外，也可用來做為函授課程的教材——我另外編了一套卡式錄音帶，供函授班學生使用。

這套教材共包括四個單元，每個單元八課，共計卅二課。第一單元是基礎程式語言，第二單元是計算機系統，第三單元是科學工程用程式語言 FORTRAN，第四單元是商用程式語言 COBOL。每課的講授時間，約為一個半小時。如果每週上課三小時，這套教材可以在一學期（約十六週）教完。教材的設計，是強調各種程式語言的共通特性，介紹層次程式設計的主要觀念，然後說明各種程式語言的主要特色。因此，這套教材應該和各程式語言的專書配合使用。各程式語言的細部問題，可以參考專書。

這套教材，首批印成專書的，包括以下三本：

CS1 認識電腦(美國知識系統學院軟體工程學教材第一單元)

CS2.1 電腦系統(美國知識系統學院軟體工程學教材第二單元)

CS3 商用電腦(美國知識系統學院軟體工程學教材第四單元)

另外還有一本講義，正在修改中，不久也將印成專書：

CS2.2 工程電腦(美國知識系統學院軟體工程學教材第三單元)

知識系統學院的學生，包括搬運工、女秘書、醫生、會計師、小學教員、中學校長……真是達到了「有教無類」的理想。因此，這套教材的設計，是給完全沒有計算機知識的門外漢使用，說明和解釋儘可能淺顯而明白易曉。但學生一旦掌握住基本觀念後，更複雜的程式設計問題，也該可以迎刃而解了。

張系國

一九八二年一月一日

認識電腦

程式設計概論及基礎程式語言

目次

序	
第一課	程式設計簡介..... 1
	附錄計算機、終端機簡介
第二課	重複資料處理..... 23
第三課	抉擇資料處理..... 39
第四課	資料結構簡介..... 57
第五課	子程式的應用..... 73
第六課	函數與變數類..... 89
第七課	檔案處理簡介..... 107
第八課	報告程式設計..... 121

第一課

程式設計簡介

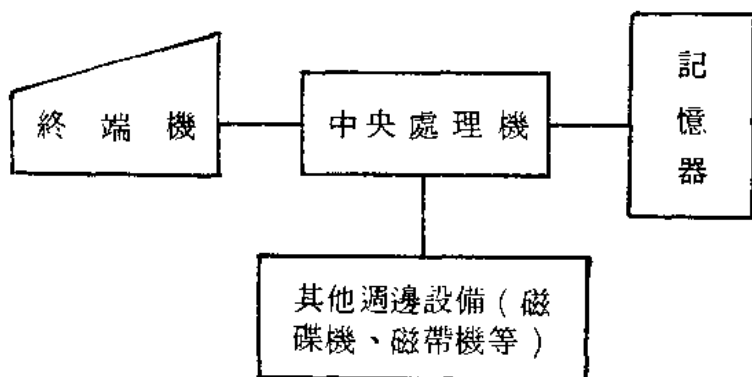
1. 電子計算機 (ELECTRONIC COMPUTER)，俗稱電腦，是近代工商業不可缺少的資訊處理 (INFORMATION PROCESSING) 及資料處理 (DATA PROCESSING) 的工具。計算機包括幾個重要部份：

(一) 中央處理機 (CENTRAL PROCESSING UNIT 簡稱爲 CPU)

(二) 記憶器 (MEMORY UNIT)

(三) 週邊設備 (PERIPHERAL DEVICES)

中央處理機等於我們人腦，它能夠遵照指令 (COMMANDS) 從事計算，輸入，輸出等工作。記憶器等於人腦的記憶部分，它能夠記下指令及資料。週邊設備等於人的眼、耳朵及口，它主要用來和外界交通。我們可將計算機的重要部份，圖示如下：



終端機 (TERMINAL) 是最常見的週邊設備，它用來輸入、輸出資料，使我們可以和計算機交談。終端機一般包括兩部份：

(一) 鍵盤 (KEYBOARD) ：類似打字機鍵盤，我們可以用鍵盤打入資料，輸送入計算機。

(二) 印字機 (PRINTER) ：計算機輸出的資料，會自動從印字機打出來。我們用鍵盤打入資料，也可以從印字機打出來。

新式的終端機，有時候不用印字機，而以顯示器 (DISPLAY) 或稱 (CRT DISPLAY) 代替。顯示器類似電視螢幕。我們打入，或計算機輸出資料，就在螢幕上直接顯示出來。這樣速度比一般印字機快速，而且也不會浪費紙張。

其他的週邊設備，包括磁碟 (DISK) 機及磁帶 (TAPE) 機等，以後再介紹。現在我們先談程式設計 (PROGRAMMING) 的基本步驟。

§ 溫習：電子計算機有那幾個重要組成部份？

2 剛才提過，計算機等於人腦，能遵照我們的指示辦事。我們如何要計算機執行一樁工作？簡單的說，我們必須使用計算機能懂的語言，對它下達指示。程式語言 (PROGRAMMING LANGUAGE) 就是計算機能懂的語言。常用的程式語言有許多種，它包括 BASIC (基礎程式語言) FORT-RAN (科學計算用程式語言)、及 COBOL (商用程式語言) 等。

以下我們要學習的，是 BASIC 程式語言的一些基本概念。

BASIC 程式語言，包括各種指令 (COMMANDS)，每種指令，教計算機從事一項工作。我們試以輸出指令 (OUTPUT COMMAND) 爲例：

〔例一〕 PRINT 12 × 3

此一指令，要計算機①計算 12×3 ，然後②印出結果。計算機執行指令，便會回答

36

〔例二〕 PRINT “ HELLO ”

此一指令的意思，是要計算機印出文數碼 “ HELLO ”。我們將 HELLO 這字串 (STRING) 放在引號 (“ 的符號) 之內，計算機就知道，應該將 HELLO 印出來。所以計算機執行指令，便會回答

HELLO

〔例三〕 PRINT “ 12 × 3 ”

請注意：凡是放在引號之內的文章碼 (ALPHANUMERIC SYMBOL)，計算機就不再去計算，祇是原封不動，依原樣印出。所以計算機執行上述指令，便會回答

12 × 3

§ 溫習：“ 12 × 3 ” 和 12×3 有何不同？

答案：前者是文章碼的字串，後者可計算而得 36 的結果。

3. 剛才提過，計算機會遵照我們的指示辦事，我們如果在鍵盤打入指令

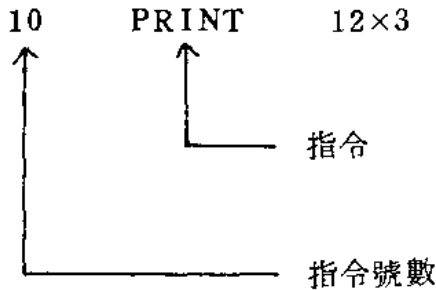
PRINT 12 × 3

計算機就會回答

36

但如果我們每次要使用計算機，都必須重新打入指令，未免太費事了。幸好計算機有記憶器，我們可以要求計算機記得我們給它的指令，下次就不必重打。

如何要計算機記得指令呢？在 BASIC 程式語言裏，我們的辦法是在指令前面，加上一個數目字：



計算機看到指令號數，就知道將指令放在它的記憶器裏，不會再忘記。但是它却並不立即執行這指令。一定要等到我們說RUN（跑！）之後，它才會去做。

〔例四〕以下的例子十分重要，請仔細看。

我們說： PRINT "HELLO"
計算機回答：
我們說： 10 PRINT "HELLO"
計算機回答：
我們再說： RUN
計算機回答：

§ 溫習：PRINT "HELLO" 和 10 PRINT "HELLO" 有何不同？

指令號數 (STATEMENT NUMBER) 不僅僅告訴計算機，必須記住指令，同時它也會依照指令號數的小到大順序，排列指令。所以我們可以打入許多指令號數不同的

· 程式設計概論及基礎程式語言 ·

指令，要計算機都存入記憶器。

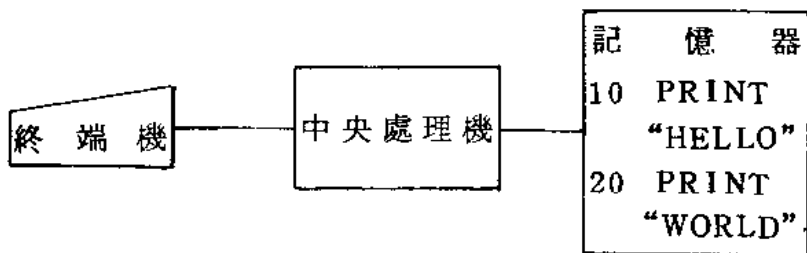
```
〔例五〕我們說： 10 PRINT "HELLO"  
                  20 PRINT "WORLD"
```

計算機回答：

我們再說： RUN

計算機回答：

請注意在〔例五〕中，我們送進去兩個指令，計算機便將它們依照順序存入記憶器。所以在我們打完兩個指令後，計算機的記憶器內有：



然後我們說 RUN(要計算機“跑”程式)，計算機便依照指令號碼的順序，執行程式。

以上的觀念，十分重要！計算機一定是照指令號碼順序，執行程式，不會亂跑。

```
〔例六〕我們說： 1 PRINT "3 × 12 = "  
                  2 PRINT 3 × 12  
或者說： 2 PRINT 3 × 12  
          1 PRINT "3 × 12 ="
```

計算機都會將以上兩個指令，存放在記憶器內，依序排列如下：

1 PRINT "3 × 12 = "

2 PRINT 3 × 12

我們再說： RUN

計算機回答：

3 × 12 =

36

§ 溫習：① "3 × 12"和3 × 12 有何不同？

② RUN 是要計算機做什麼？

③ 指令號數有那兩種功用？

4. 上面我們介紹了指令是什麼？如何將指令存入計算機的記憶器？如何要計算機執行存在記憶器內的指令？

我們存在記憶器內的指令，就叫做程式 (PROGRAM)。所以，程式由許多指令構成。例如〔例六〕的簡單程式，有兩個指令。

更複雜的程式，有數百、數千、甚至上萬個指令！

我們如何檢查計算機的記憶器內有何程式呢？

我們可以說：LIST

LIST 的意思，是要求計算機將現存放在記憶器內的程式，依序列出。用中文說，就是印程式。RUN 的意思，就是跑程式。

所以，程式設計的基本步驟如下：

(→) 將指令放入計算機的記憶器；

· 程式設計概論及基礎程式語言 ·

(⇒) 印程式 (LIST) ;

(⇒) 跑程式 (RUN) ;

上述過程如果有錯誤，就修改程式，再重複(⇒)，(⇒)的工作。

〔例七〕我們想設計一個程式，印出年薪 (YEARLY SALARY) 的字樣，再印出年薪 15000；然後再印出月薪 (MONTHLY SALARY) 的字樣，再印出月薪的數目。

(第一步) 我們將指令放入計算機的記憶器；

```
10 PRINT "YEARLY SALARY"  
20 PRINT 15000  
30 PRINT "MONTHLY SALARY"  
40 PRINT 15000 ÷ 12
```

(第二步) 我們印出程式，以便檢查：LIST

(第三步) 我們要求計算機執行程式：RUN

上面過程，如果有錯誤，我們就修改程式。

§ 溫習：在〔例七〕我們說RUN 之後，計算機回答如何？

答案：

YEARLY SALARY
15000
MONTHLY SALARY
1250

5 以上〔例七〕的程式，可以印出年薪及月薪，但是如果年薪不是15000，這程式就沒用了！

如何設計程式，能夠印出不同的年薪和月薪呢？我們可以想到，計算機既然能夠儲存指令，它也可以儲存別的資料吧？

如何儲存資料入計算機的記憶器？

我們可以用不同的名稱來代表不同的資料。例如年薪，我們可以用 S 來表示。S 是一個變數 (VARIABLE) 的名稱。在計算機的記憶器內，我們就留有一個位置，用來放置 S 的值。

§ 溫習：記憶器內的指令，都有指令號碼。記憶器內的資料，都有特別的名稱 (VARIABLE NAME)。

〔例八〕我們修改〔例七〕的程式，將年薪換成變數 S，就得到如下的程式：

```
10 PRINT "YEARLY SALARY"  
20 PRINT S  
30 PRINT "MONTHLY SALARY"  
40 PRINT S ÷ 12
```

〔例八〕的程式比較有用。如果 S 的值不同，印出的年薪和月薪就不一樣。

例如 S 的值是 12000，我們說 RUN 之後，計算機回答是：

```
YEARLY SALARY
12000
MONTHLY SALARY
1000
```

§ 溫習：如果〔例八〕程式，我們修改其中的第 40 號指令如下：

```
40 PRINT S ÷ 9
```

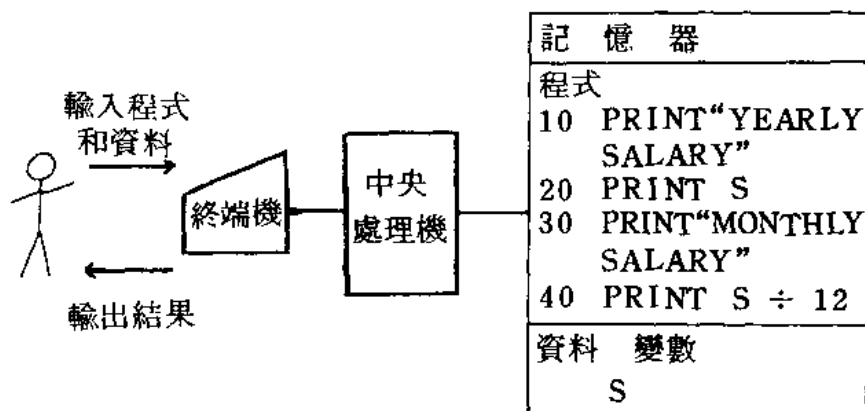
如果 S 的值是 18000，我們說 RUN 之後，計算機的回答是什麼？

答案：

```
YEARLY SALARY
18000
MONTHLY SALARY
2000
```

變數的名稱，在基礎程式語言裏，一般用一個字母來代表。在高級基礎程式語言裏，用一個字母、兩個字母、或一個字母跟著一個數字來表示，例如 S、N1、B5……都是合適的變數名稱；7、6B，……則不能用來做為變數名稱。

6. 上面我們介紹了，如何在記憶器內儲存指令，如何用變數來表示資料。〔例八〕的程式和資料變數，就如下圖所示。



變數 (VARIABLE) 等於是個空油瓶，資料等於是酒。如何將資料 (酒) 放入變數 (酒瓶) 之中？

我們可以使用輸入指令 (INPUT COMMAND)，讓計算機能夠輸入適當的資料。

〔例九〕在〔例八〕的程式中，我們再增加一條輸入指令：

```
5 INPUT S
```

計算機在執行這輸入指令時，就會要求我們送入一個數目，它把這數目放入 S 裏面，通常它會打出一個問號，等我們輸入資料。

我們的完整程式如下：

```
5 INPUT S  
10 PRINT "YEARLY SALARY"  
20 PRINT S  
30 PRINT "MONTHLY SALARY"  
40 PRINT S ÷ 12
```