

6週COBOL

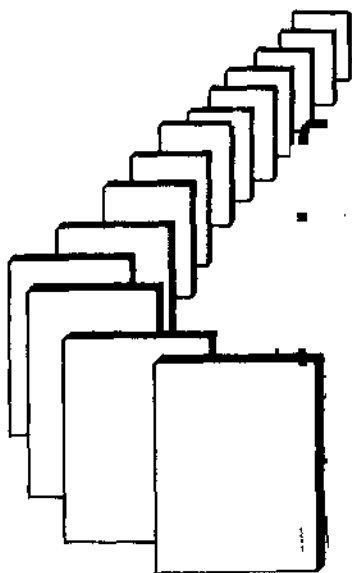
程式入門

陳能傑編著



6週COBOL程式入門

陳能傑 編著



6週COBOL程式入門

作者：陳能傑

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

總經理：宏基書店有限公司

地址：香港皇后大道一〇五號

印刷者：金源印刷廠

地址：香港蘇杭街十九號

定價：港幣拾玖元

有著作權・翻印必究

H.K.\$19.00

序

總統對資訊工業的訓示：「我國科技要趕上工業先進國家，必須普及電腦教育及使用，由政府、教育界及企業界共同努力，全力發展電腦及資訊工業，以促進工業的升級，增強我國產品在國際市場上的競爭力。」

然而，於資訊工業進步神速的今日，學電子計算機的人亦與日俱增，學電腦已不再是大學生的專利了，而 COBOL 更是電腦中最通用的語言之一。有鑑於此，作者乃利用公餘之暇，特將數年教學心得及實際工作經驗，精心編撰而成此書。

本書的最大特色，乃係以最淺顯的敘述、深入淺出、循序介紹，並儘量以圖形、流程圖配合實例反覆詳盡說明，期使初學者能一目瞭然，融會貫通，而能啓發學習者的興趣，此乃作者最大的願望。

本書的設計是針對高商、高工、商專、工專、電腦進修班、未具電腦知識的初學者及企業界卻學 COBOL 者研讀。俗語說「工欲善其事，必先利其器」更願此書的問世，能帶給電腦界有所助益。

本書之編成，承蒙資訊教育推廣中心的周主任家榮的鼎力支持以鼓勵，劉小姐、林惠芳小姐的仔細校稿在此一併致謝。惟作者才疏學淺且倉促付梓，謬誤之處，尚祈海內先進不吝教正，俾再版時予以更正。

陳 能 傑 謹識

目 錄

第一章 COBOL 語言簡介

1-1 電腦程式概念	1
1-2 COBOL 語言的特性	5
1-3 COBOL 語言的基本結構	7
1-4 COBOL 語言的字符與單字	10
1-5 資料結構	16
1-6 資料的種類	24
1-7 簡易的 COBOL 程式範例	34

第二章 識別部與設備部

2-1 COBOL 編寫紙的使用方法	39
2-2 識別部	46
2-3 識別部的實例研討	50
2-4 設備部	51
2-5 機型節（可有可無）	52
2-6 輸入輸出節	56
2-7 設備部的實例研討	60

第三章 資料部

3-1 檔案節	67
3-1-1 檔案描述句	68
3-1-2 記錄描述句	77

3-2	工作暫存節	86
3-2-1	獨立項目的用法	87
3-2-2	集體項目的用法	87
3-3	資料部之實例	90

第四章 程序部

4-1	OPEN敘述	96
4-2	READ敘述	99
4-3	PERFORM..... UNTIL 敘述	101
4-4	MOVE敘述	106
4-5	WRITE	119
4-6	CLOSE 與 STOP/ RUN 敘述	122
4-7	COBOL 程式範例	124

第五章 算術運算指令

5-1	ADD敘述	133
5-2	SUBTRACT敘述	139
5-3	MULTIPLY敘述	144
5-4	DIVIDE敘述	147
5-5	ROUNDED 和 ON SIZE ERROR兩取捨子句	153
5-6	COMPUTZ敘述	158
5-7	COBOL 程式範例	168

第六章 條件敘述與DISPLAY敘述

6-1	關係條件	175
6-2	正負號條件	181
6-3	種類條件	182
6-4	複合條件	184
6-5	簡單的PERFORM敘述	191

6-6	PERFORM..... UNTIL 條件敘述	194
6-7	DISPLAY 敘述	197

第一章 COBOL語言簡介

1-1 電腦程式概念

程式 (PROGRAM) 是由一連串具有邏輯順序的指令 (INSTRUCTION) 所構成，其目的在處理輸入的資料並將產生的結果輸出。簡單地說，為了解決某項工作（如計算學生的學期成績）並藉助電腦來達成，我們必須寫出能指示電腦完成該項工作的一些指令，這些指令羣可稱為“程式”。

指令 (INSTRUCTION) —— 程式是由指令所組成，故指令是程式的基本單元。每一指令命令電腦做一件事，猶如在軍中長官下達的命令一樣，此命令是命令部屬做事，而由程式設計師寫出的指令是命令電腦來執行這些指令。指示電腦做事的命令稱為指令，指示部屬做事的命令稱為口令。每一部電腦都有其固定的指令集 (Instruction Set)。一般而言，大型電腦的指令種類較迷你電腦來得多，指令種類一多，越容易解決問題，方法亦較簡單。

我們都知道，電腦只懂得機器語言 (Machine Language)，這種語言由 0 與 1 組合而成。亦就是電腦只能執行機器語言之指令，可是要編寫這類型的指令是一件冗長而繁雜的工作，寫程式者（亦可稱為程式設計師）必須記住記憶體位址 (Memory Address) 與使用複雜的操作碼 (Operation Code)。

由於用機器語言寫程式是相當困難的，故電腦專家發展程式設計技術，使程式設計師 (Programmer) 可用類似英語的指令編寫程式。但是在電腦要執行這種指令之前必須先將之翻譯或編譯 (Compile) 成機器語言，而這翻譯的工作是經由電腦本身利用一些控制卡或命令 (Command) 來完成的。

電腦在商業應用方面已日趨普遍，然而在商業方面的業務往往是資料數量龐大，計算簡單，它需要有較高速率的輸入與輸出設備，方足以處理龐大的業務資料。故

2 6週COBOL程式入門

其與處理科技用途的電子計算機之性能不同（它是用來處理資料簡單，計算複雜的問題，此種以FORTRAN語言較為適宜）。有鑑於此，美國聯邦政府乃邀集主要電腦廠商、客戶及學術機構等，于西元1959年聯合組成了資料系統語言委員會（Conference On Data System Language，簡稱CODASYL），共同研訂開發一種適合於各種廠牌電腦使用之標準化程式語言，此即所謂的商用程式語言（Common Business Oriented Language，簡稱COBOL）。

COBOL除適用於各種廠牌電腦外，其語法或句型結構均與英文相類似，極易閱讀與瞭解，而且容易學習，在短期內即可完成訓練，故普受一般人所樂用，目前國內各企業界或公司行號，大都使用COBOL作為電子資料處理之工具。

COBOL與FORTRAN一樣，均是一種高位準程式語言（High Level Language）。若想執行COBOL程式，必須由COBOL編譯程式（Compiler）將之翻譯成機器語言。如下所示：



由上圖可知程式設計師以一種高位準語言寫成的一連串指令，稱之為原始程式（Source Program）。電腦不能直接執行此原始程式，必須先將其翻譯成機器語言，電腦才能執行此程式。通常我們都用打卡機將寫好的原始程式打成卡片，並將此原始程式卡（Source Deck）送入計算機（或直接經由終端機（Terminal）送入電腦），然後執行編譯程式（Compiler），將此原始程式編譯成機器語言（此機器語言稱之為目的程式（Object Program）），爾後計算機才能執行此程式。一般來說，針對一種程式語言所寫成的專用程式，其能將原始程式翻譯成目的程式。此種專用程式稱之為編譯程式（Compiler）。例如，COBOL之COBOL編譯程式（COBOL Compiler）、FORTRAN的FORTRAN編譯程式（FORTRAN Compiler）…等等。

接着將說明一個程式從設計至完成所涉及之步驟（如圖1-1）為：

1. 由程式設計師依據解決該問題的方法，用COBOL語言編寫程式。
2. 將寫好的程式經由打卡機將之打成卡片。

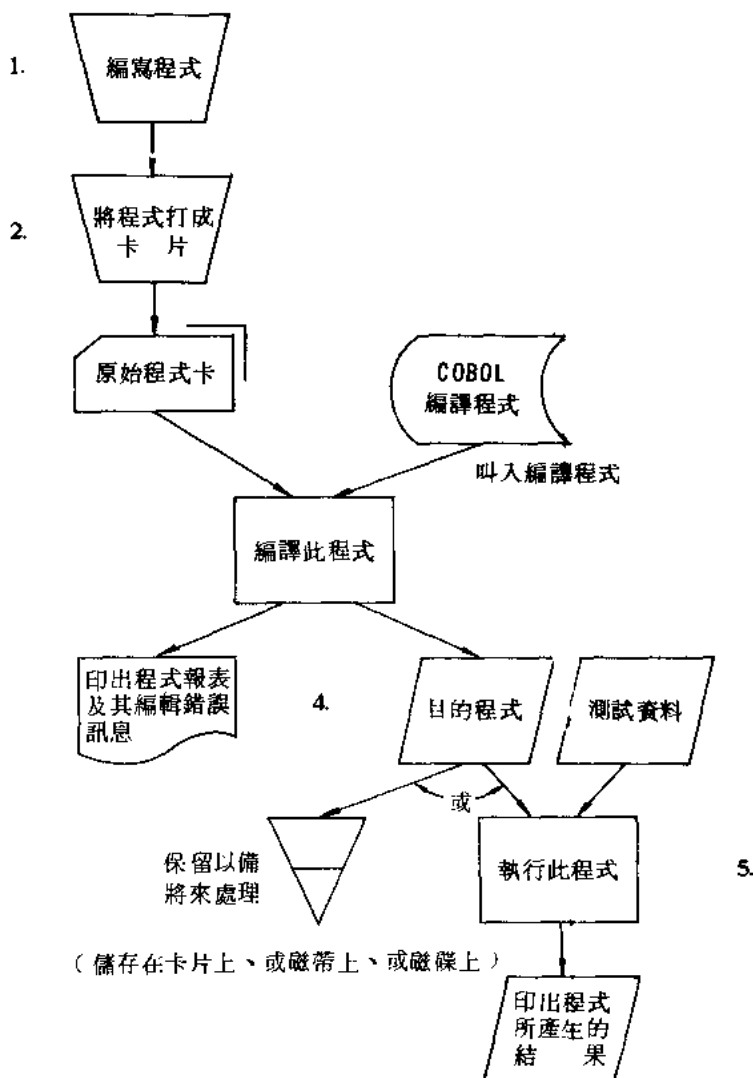


圖 1-1 程式設計所涉及之步驟

4 6 週 COBOL 程式入門

3. 將原始程式卡送入計算機中，並叫入編譯程式來編譯該程式。執行編譯後，若無編譯錯誤 (Compile Error) 則產生目的程式；若有編譯錯誤則印出程式報表及編譯錯誤訊息。更正錯誤，並重新編譯直到完全正確時為止。
4. 若目前不需要執行該程式，可將目的程式打在卡片上 (這些卡片稱之為 **Object Deck**) 或儲存於其它媒體上 (如儲存在磁帶或磁碟上)。以後我們將不需要再編譯此原始程式即可直接執行該目的程式。
5. 讀入輸入資料，執行此程式並輸出程式所產生的結果。

當計算機編譯一個原始程式時，編譯程式會將不合語法之錯誤列印出來，以便程式設計師根據這些錯誤訊息 (Error Message) 加以更正，並重新編譯直到其完全正確時為止。一般而言，在編譯時將會發現任何觸犯程式設計規則的錯誤，但無法知道此程式之邏輯是否正確。然而編譯錯誤有些是屬於語法錯誤 (Syntax Error)，如在寫程式時將代表加法運算的 ADD 指令寫成 AD，經編譯後計算機將印出一錯誤訊息。在編譯一個原始程式時，若發現嚴重的錯誤時，則計算機將會停止編譯。

必須強調的一個觀念是編譯錯誤 (Compile Error) 與邏輯錯誤 (Logic Error) 是不同的。一個程式的邏輯順序不對時，在執行此程式時 (假設此程式已無編譯錯誤)，才能發現邏輯錯誤。亦即在編譯時的錯誤是屬於編譯上的錯誤，在執行程式時才會有邏輯錯誤的發生。通常計算機是無法判斷程式之邏輯是否錯誤，程式設計師可依其工作經驗及一些查錯方法方能找出此種錯誤。

〔自我測驗〕

1. 程式是一連串具有邏輯順序的_____組合而成。

☐ 指令

2. 若要計算機能執行某程式，此程式之所有指令都應該是_____語言的形式。

☐ 機器

3. 不是用機器語言寫成之程式在執行之前必須_____。

☐ 翻譯或編譯成機器語言

4. 將 COBOL 程式轉換成機器語言的程序稱為_____。

☐ 編譯

5. 用高位準語言（如 COBOL）所寫成的程式，稱為_____程式。

答 原始

6. 原始程式卡是指_____。

答 打在卡片上的程式卡

7. 已被轉換成機器語言之一連串指令，稱為_____程式。

答 目的

8. _____程式，是將_____程式轉換成_____程式。

答 編譯，原始，目的

9. 編譯時所發現的錯誤即表示_____，此錯誤通稱_____錯誤。

答 在使用符號設計程式時，違犯程式設計規則
語法

10. 程式之邏輯僅能經由_____核對之。

答 測試或執行程式本身

11. 將程式編譯成目的程式之後，即可_____此目的程式或_____。

答 執行

儲存以供未來使用

12. COBOL 是_____之縮寫。

答 COmmon Business Oriented Language

13. COBOL 是一通用語言因為_____。

答 可用於許多不同廠牌，不同機型的計算機

14. 通常 COBOL（適合，不適合）用於科學應用。

答 不適合

1-2 COBOL 語言的特性

COBOL 語言的特性為：

COBOL 語言的特性

1. COBOL 是目前最被廣泛採用的商用程式語言。
2. COBOL 是許多計算機所通用的一種計算機語言。
3. COBOL 具有普及性。
4. COBOL 具有持久性。
5. COBOL 是一類似英語的語言，容易學習。
6. COBOL 程式較易編寫與閱讀。

一、COBOL 是目前最被廣泛採用的商用程式語言

電子計算機在商業上的運用已日趨廣泛，但因商業上的業務往往是資料數量龐大而計算簡易，因而需要較高速率的輸入與輸出設備，方能處理此龐大的業務資料，於是便發展出 COBOL 語言。因此，它是專為配合商業應用而精心設計的一種計算機語言。

二、COBOL 是許多計算機所通用的一種計算機語言

大多數計算機製造商都設計其電腦所能接受的 COBOL 編譯程式，因此同一 COBOL 程式只要稍加修改，就能在許多不同的計算機（如 IBM/370 和 UNIVAC 1100）上編譯成電腦的計算機語言。

三、COBOL 具有普及性

各商業應用系統，若採用相同的程式語言，公司可自由選擇在不同廠牌的計算機上執行，因而使計算機使用者將具更大的彈性。

四、COBOL 具有持久性

從舊計算機轉換至較好、較新的計算機時，也不會產生重大的難題，而未來的計算機亦將保有 COBOL 的功能。

五、COBOL 是一類似英語的語言且容易學習

COBOL 的結構有點類似英語的結構，很容易懂，而它所用的語言即是一般英語用的語句，所以很容易學習。且 COBOL 程式的撰寫也較其他語言簡單，因此 COBOL 程式訓練可在很短的時間內完成。

六、COBOL 程式較易編寫與閱讀

由於 COBOL 具有與英語相類似的特質，故其程式較易編寫與閱讀。

〔自我測驗〕

1. COBOL 的特色有那些？

- 答**
1. COBOL 是目前最被廣泛採用的商用程式語言。
 2. COBOL 是許多計算機所通用的一種計算機語言。
 3. COBOL 具有普及性。
 4. COBOL 具有持久性。
 5. COBOL 是一類似英語的語言且容易學習。
 6. COBOL 程式較易編寫與閱讀。

1-3 COBOL 語言的基本結構

COBOL 語言之結構與一般文章結構相同，其內容結構大致分為部、節、段與句子等。其

每部 (DIVISION) 分為若干節 (SECTION)；

每節分成若干段 (Paragraph)；

每段則由一個或一個以上的句子 (Sentence 或 Entry) 組成；

句子又由子句 (Clause) 或敘述 (Statement) 組成；

而子句或敘述均由單字 (Words) 構成；

單字則由一個或數個字符 (Characters) 合併而成。(請參閱圖 1-2)

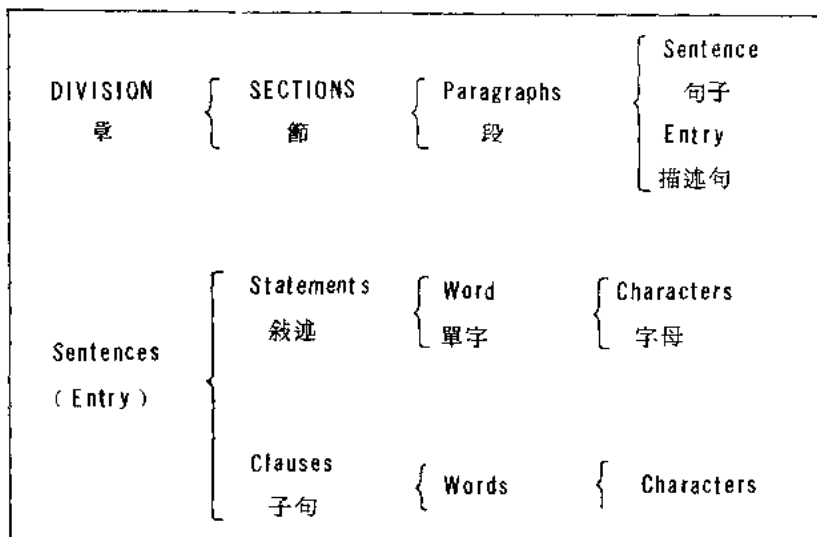


圖 1-2 COBOL 程式之語法結構

每一 COBOL 程式均由四個部 (Divisions) 組成。各部均以類似英語的 (English-like) 方式寫成，以減輕程式設計工作並使非資料處理人員容易了解程式。茲將四大部所具有的特定功能詳述如下：

COBOL 四大部	
名 稱	目 的
IDENTIFICATION DIVISION (識別部)	使計算機識別程式，亦即用以設定程式名稱。並提供文件有助於非資料處理人員分析或閱讀程式。
ENVIRONMENT DIVISION (設備部)	描述特定程式所用到的計算機設備。
DATA DIVISION (資料部)	描述程式將處理之輸入與輸出區域。並定義資料處理所需之常數或工作區。
PROCEDURE DIVISION (程序部)	包含讀入與處理資料並產生輸出所需之實際指令。

IDENTIFICATION DIVISION.	(識別部)
PROGRAM- ID.	(第一段)
AUTHOR.	(第二段)
INSTALLATION.	(第三段)
DATE-WRITTEN:	(第四段)
SECURITY.	(第五段)
REMARKS.	(第六段)
ENVIRONMENT DIVISION.	(設備部)
CONFIGURATION SECTION.	(機型節)
SOURCE-COMPUTER.	(第一段)
OBJECT-COMPUTER.	(第二段)
SPECIAL-NAMES.	(第三段)
INPUT-OUTPUT SECTION.	(輸入、輸出節)
FILE-CONTROL.	(第一段)
I-O-CONTROL.	(第二段)
DATA DIVISION.	(資料部)
FILE SECTION.	(資料構節)
.....	
.....	
WORKING-STORAGE SECTION.	(工作暫存節)
.....	
.....	
PROCEDURE DIVISION.	(程序部)
READ-RTN.	(第一段)
.....	
.....	
CLOSE-RTN.	(第二段)
.....	
.....	
CALC-RTN.	(第 n 段)
.....	

圖 1-3 COBOL 程式的架構與其先後次序

每一 COBOL 程式均可分為：

- 一、識別部 (IDENTIFICATION DIVISION)，
- 二、設備部 (ENVIRONMENT DIVISION)，
- 三、資料部 (DATA DIVISION)，
- 四、程序部 (PROCEDURE DIVISION) 等四大部。

識別部內容較為簡單故不分節，僅分成數段。設備部分成機型節 (CONFIGURATION SECTION) 與輸入輸出節 (INPUT-OUTPUT SECTION)，前者分為 SOURCE-COMPUTER，OBJECT-COMPUTER 及 SPECIAL-NAMES 等三段，後者分為 FILE-CONTROL 及 I-O-CONTROL 兩段。資料部分為資料檔節 (FILE SECTION) 及工作暫存器 (WORKING-STORAGE SECTION) 兩節，節以下則不分段。而程序部除特殊情形外可不分節，部以下即直接分成若干段。

1-4 COBOL語言的字符與單字

[COBOL語言的字符]

無論學習那一種電腦程式語言，都必須先知道那種語言的語法，就如同學習英語一樣，首先要知道有那些字可用，以及文法如何，這樣一來，別人才聽得懂你說的英語，否則別人是無法了解的。同理，我們在學 COBOL 語言之前，也要了解它的規則，否則計算機就不能編譯你的 COBOL 程式，亦無法執行 COBOL 程式了。然而，在 COBOL 程式語言中，所用的字符通常包括文字、數字與特殊字符這三類，計有 51 個。茲歸類分述如下：

1. 英文字母 (Letters)：A ~ Z 等 26 個字符。
2. 數字 (Digits)：0, 1, 2, 3, 4, 5, 6, 7, 8, 9 等 10 個字符。
3. 特殊字符 (Special Characters)：如下表所示。