

INTRODUCTION TO AMERICAN
NATIONAL STANDARD

COBOL

商用電腦程式進階

William M. Fuori 原著



本書根據美國國家標準詳列各式練習，
運用程式語言有正確答案方便自作檢查

原 序

何謂 COBOL ?

在 1950 年代早期工商業界首次引進電腦的技術，對於標準商業化程式語言的需求日趨迫切，爲了保證此種語言能夠國際化，一個由計算機製造廠商，美國政府商業人士，以及企業界的使用者聯合組成的資料系統語言委員會 (Conference on data System Language 簡稱 CODASYL) 研商發展一套標準商業化程式語言，在 1959 年委員會發展出通用商業程式語言 (Common Business Oriented Language)，簡稱 COBOL，此套程式語言通用於不同的電子計算機製造廠商的機器，由於語言本身的優點爲美國政府及企業界推崇備至。

在 1968 年美國國家標準協會 (American National Standards Institute 簡稱 ANSI) 出版 COBOL 程式語言通則，爲各有興趣團體代表所一致接受，在 1974 年 ANSI 更出版修訂過的 COBOL 標準，這是自 1968 年出版的標準 COBOL 語言以來首次結合電子計算機製造廠商及使用者經驗精華所發行的刊物。

指令特性—本書所有的指令格式均取自直接授課的經驗，以及許多專家、學者、商業人士對 COBOL 指令的許多建議，以美國國家標準通用商業程式語言爲藍本，可能較時下一般 COBOL 教材爲新穎和具建設性，對於以往 COBOL 具抽象的指令能夠提供給學習者實際的體認，及嶄新的學習趨向。

標準化的 COBOL—本書介紹主要 COBOL 語言指述廣泛的討論，提供學習者及程式設計師，由 1968 年至 1974 年期間不斷修訂的 COBOL 新觀念，在附錄 E 尚有一般指令的格式通則摘錄。

直截了當的介紹—本書所謂直截了當的介紹有兩種意義：第一能夠使讀者容易瞭解 COBOL 程式語言指述的用法，第二及早介紹給學習者 COBOL 語言基本的需求及程序，例如：介紹給學習者 ACCEPT (讀資料) 和 DISPLAY (寫資料) 述句 (在第三章)，此種及早的介紹可使學習者能夠立即編寫簡單完整的程式，提高學習興趣。

一旦學者能夠編寫簡單及完整的程式，自然會產生無比的信心，更能接受越來越複雜困難的其它 COBOL 指令，以資料部爲例，邊學邊做，學生由簡易到複雜逐步的學習。

在學習完基本的 COBOL 觀念，必須接受詳細的整體性的程式設計經驗，諸如表格處理，磁碟操作處理和排序特性等。

足夠的練習教材—本書練習教材區分兩大部份，第一種是每節後面的練習題，主要的目的在每節學習後即刻強迫再灌輸本節必要的描述觀念，增強印象。

第二種是在每章後面有複習題，包括是非題、選擇填充題、問答題及解釋名詞都是本章重點精華的總復習，讓學者在本章學習完畢後有第二次加深觀念培養的機會。

在每章後面複習題的問答題，讓學者根據自己累積的經驗、實際體驗企業界的各種狀況，例如讓學生編寫某公司銷售員每人銷售績效日報，或航空公司旅客訂位計劃等，這些都是學者日後在企業界所必須面臨的問題。

本書所有程式範例及作業都經過程式編譯，正式上機操作過，俾確定其正確無誤，再將程式列出作為範例，並且依據操作手冊安排設計的作業。

區分不同製造廠商 COBOL 指令—在本書 COBOL 指令介紹時，以 IBM 電腦系列的指令為藍本，凡是 IBM 電腦系列所專用的指令都經特別強調，如果學者和程式設計師是使用 IBM 的機器，則更易於瞭解，而非使用 IBM 機器的學者或程式設計師也能輕易分辨出不同製造廠商之間指令的區別，有所概念，本書偏重於 1968 年至 1974 年間 COBOL 一般指令格式的介紹，給各位一個比較性的參考。

適用在儲存量大的系統—自從發展出虛擬儲存體 (Virtual Storage)，目前電腦儲存量設定愈來愈大，多數程式設計師編寫的 COBOL 程式可在 VS/DOS (Virtual Storage/disk operating System) 或 VS/OS (Virtual storage/operating System) 內作業，基於此，有必要讓學者熟悉 VS/DOS 或 VS/OS 的作業情形，有關於虛擬儲存體設備的知識請學者參閱 IBM 公司所出版一系列相關的書籍即可了解。

可選擇性的教材—本書提供一些可選擇性的教材，遍佈在章節，問題和習題內，以“*”標示出，如果學者有興趣可以深入研究，作為參考。

本書結構及內容：

本書區分四個單元介紹美國國家標準 COBOL 程式語言，共計十四章，第五單元更有從附錄 A 至附錄 E 的補充材料。

第一單元（第一章和第二章）介紹美國國家標準 COBOL 程式語言概論及基本原則讓學者瞭解 COBOL 程式語言的由來，如何漸次發展而成，以及 COBOL

編譯程式的功能（轉換 COBOL 語言述句成為機器可以執行的識別碼）。

第二單元（第三章至第七章）介紹學者 COBOL 設計的程序（問題分析，發展適當的運算及流程圖，編寫 COBOL 指令解決問題），第一個 COBOL 程式著重由讀卡機輸入卡片資料及經由印表機列出運算的結果程式內討論 ACCEPT 指述與 DISPLAY 指述的用法，一旦學者熟悉此種類型的程式編寫，即可開始研究基本的算術運算指述，設計資料檔案，第二單元結束後，所設計的作業更能加深 COBOL 的重要觀念。

第三單元（第八章至第九章）著重於 COBOL 分歧指令和迴圈處理的介紹，順序討論 IF, GO TO, ALTER 及 PERFORM 指述的用法，和第三單元相同，此單元的作業在加強 COBOL 的基本設計觀念。

第四單元（第十章至第十四章）介紹學者一些 COBOL 觀念的進階，及特殊性能包括表格的處理，磁帶檔案的設計，磁碟儲存觀念，磁碟檔案的設計，以及 COBOL 排序特性討論，並且列舉為數不少的例題加以說明。

第五單元（附錄 A 至附錄 E）介紹一些操作方面的補充材料包括 COBOL 程式除錯介紹（附錄 A），COBOL 程式的文件說明記載方式（附錄 B），OS 任務控制語言介紹（附錄 C），每一個附錄都包含商業範例的說明及詳細的註釋，此外更包括美國國家標準 COBOL 字彙辭典（附錄 D），主要美國國家標準 COBOL 一般性格式通則（附錄 E），此格式通則由 1968 年至 1974 年之間所有的格式並列在一起。

由於本書的目的並非專注於個別指令的描述，而是強調一系列的程式設計的技巧，因此各章節之間彼此都有關連性，在每一個單元內，更是整體性的描述使學者有清晰的觀念，本書的編寫是著者累積教授的心得，結合資料的處理，技術及商業上的實例，匯聚的精華，相信學者能夠提高興趣，容易瞭解每一個指令的用法，熟悉程式的設計，獲得最大的效益。

教師的手冊及指導關鍵

本書亦提供教師作為指導關鍵及參考的手冊，尤其在 COBOL 課程教授方面更能提供適合學生的學習方式，每章節後面的習題都是重要觀念的複習，問答題及解釋名詞更可作為課堂上相互討論的課題。

每章後面的習題未附解答，希望學生能夠自己設法解決，照題目要求編寫出程式指令，寫出適當的輸入及輸出資料格式，也能督促學生及教師設計適當的測試資料，俾能產生合適的輸出結果。

譯 者 序

由於科學的進步，工商業的發達，目前電子計算機在國內的應用日趨廣泛。電子計算機隨製造廠商的不同，程式語言種類相當多，以IBM電腦系統言有ASSEMBLER, COBOL, FORTRAN, RPG 及 PL/I 等語言，除ASSEMBLER為低位準(low level)語言，接近於機器語言外，其餘均為較易學習的高位準(High level)語言，其中COBOL係由Common Business Oriented Language 縮寫而成，通常吾人稱為通用商業程式語言，專為處理商務資料而開發設計的程式語言，由於此種程式語言結構標準化，具有文書的性能，易於閱讀及瞭解，是以非常受工商業界的歡迎。

本書採用循序漸進的方式，配合適宜的範例，逐章解說COBOL程式語言的各種指令用法，每章更有練習題及複習題，加強基本觀念的訓練，此外添加磁帶檔案，磁碟檔案，各種檔案結構的介紹，此種軟體性的探討本屬於Data Management 的範疇，其目的在使讀者能對電腦系統有更深一層的認識。附錄內的COBOL基本除錯技巧與COBOL程式語言文件說明記載處理，都是與電腦實際作業息息相關，作者此種理論與實務相互配合對照，更能提高讀者學習的興趣，這是坊間相關書籍所不能相比擬的所在，亦是本書獨創的優點。

附錄C為OS 任務控制語言(JOB CONTROL LANGUAGE)簡介，由於目前國內一般電腦設備均是DOS系統，是以省去不譯，附錄D為COBOL語辭的英文註釋，翻譯後可能會與原意有所出入，亦決定不予刊出。此兩附錄讀者如有興趣，請參考原文書。

譯者利用公餘之暇翻譯本書，疏漏及錯誤之處在所難免，尚祈諸先進不吝指正。

米 功 偉

謹識

68.9.10.

第一單元

通用商業程式語言產生之背景

第一章 概論

目 錄

1. 通用商業語言發展簡史.....	1
2. COBOL 的優點及特徵 (COBOL ADVANTAGES AND FEATURES).....	2
3. 電腦作業化需要些什麼條件 (PROBLEMS SUITABLE FOR COMPUTERIZED SOLUTION).....	3
• 可以定義的 (DEFINABLE)	
• 可以重複的 (REPETITIVE)	
• 儲存或運算的空間 (VOLUME OF STORAGE OR CALCULATIONS)	
• 費用的衡量 (COST-JUSTIFIABLE)	
4. 由問題分析至作業說明的記載 (FROM PROBLEM ANALYSIS TO DOCUMENTATION).....	4
• 問題分析 (PROBLEM ANALYSIS	
• 作業系統流程 (FLOWCHARTING)	
• 程式編寫、除錯及執行 (CODING, DEBUGGING, AND EXECUTING THE PROBLEM)	
• 作業說明記載 (DOCUMENTATION)	
5. COBOL 程式編寫完成後的處理 (AFTER THE COBOL PROGRAM IS WRITTEN).....	9

目 錄

第一單元 通用商業程式語言產生之背景

第一章	概論.....	1
第二章	通用商業程式語言基本原則.....	15

第二單元 COBOL程式編寫簡介

第三章	編寫簡單的 COBOL 程式.....	38
第四章	算術運算.....	67
第五章	程式檔案的設計.....	108
第六章	程式部的範例.....	137
第七章	衍論.....	178

第三單元 改變程式中指令的走向

第八章	分歧指述.....	206
第九章	微調指述.....	245

第四單元 COBOL 觀念進階

第十章	表格處理.....	266
第十一章	磁帶檔案的程式設計.....	307
第十二章	直接取用儲存體資料的觀念.....	339
第十三章	DASD 的程式設計.....	365
第十四章	COBOL 程式排序特性.....	405

第五單元 補充資料

附錄 A	COBOL 程式的除錯.....	434
附錄 B	COBOL 程式的文件說明記載.....	445
附錄 E	格式通則簡介.....	456

1.1 通用商業語言發展簡史

隨著社會急遽發展，電子計算機的應用不再侷限於科學技術性的探索，更需開展如何配合商業界解決各種新穎的，錯綜複雜的問題，是以電子計算機製造廠商及有心的使用單位開始，著手研究如何以現有電子計算機的性能，在最少成本花費目標之下，從事最低限度技術性機件的轉換，程式計劃的修訂，人員的再訓練，來謀求更具效益的資料處理系統，同時發展一套包容不同廠牌電子計算機型，儲存媒體裝置的商業性通用程式語言，不但能夠使一般專業化程式設計人員在短時間內即可編寫程式，作為一種作業說明規範，並且由於它的簡易敘述性，即使是非專業性諸如會計人員、查帳人員等，亦能輕易的閱讀程式，瞭解其意義。

1959年5月由計算機製造廠商，美國聯邦政府，及其他有興趣學術團體的代表，聯合組成“資料系統語言委員會”(Conference On Data System Languages 簡稱 CODASYL)，發展一套簡易化、標準化的商業性程式語言，1960年4月以 COBOL (Common Business Oriented Languages) 為名，發表報告書，於是成立通用商業程式資料系統語言委員會，專門負責通用商業程式語言必要的修改，期能制定商業界的程式語言規範，進而推廣至國際性廣泛的使用。

基於 COBOL 程式語言在商業界運用得非常成功，顯示該程式語言擁有光明遠大的前程，有鑒於此，美國國家標準協會 (United States of America Standards Institute 簡稱 USASI) 參考 CODASYL 委員會及各製造廠商不同的規範，積極展開 COBOL 語言通則研究，終於在 1968 年 8 月制定出“美國國家標準通用商業程式語言 ANS (American National Standard) COBOL”，1974 年底在 ANS COBOL 修訂版本附錄 E 中摘錄美國 IBM 電腦公司最新 COBOL 編譯程式通則 (COS FULL ANS COBOL COMPILER)，不過由於一般 COBOL 編譯程式仍以 1974 年底 ANS COBOL 最新版本為主，是以該項附錄不擬收錄在本書。

1-1 練習題

1 在 1960 年 4 月由_____委員會發行以 COBOL 為名的報告書。

答：CODASYL (資料系統語言)

2 資料系統語言委員會乃是由許多電子計算機製造廠商，美國聯邦政府，以及其他有興趣的學術團體_____所組成。

答：代表

3 該委員會的目的是發展一套簡易化_____的商業程式語言。

答：標準化

4 目前最盛行的通用商業程式語言是_____通用商業程式語言。

答：美國標準

1.2 COBOL 的優點及特徵

COBOL 程式語言是專為商業或與商業有關業務設計的一種易學易讀的高位準 (High-Level) 程式語言，其具體的特徵如下：

- 1 COBOL 程式係由精確拼音的英文單字及短句組成。
- 2 適合於不同廠牌電子計算機及各型輸入輸出裝置使用。
- 3 在不同廠牌電子計算機中使用 COBOL 程式語言，僅有少量的差異。
- 4 COBOL 程式乃是使用一般商業性的術語編寫，是以如會計人員，查帳人員，甚至決策階層者等非專業人員，祇要是具有少許電腦基本概念，即能輕易的瞭解。
- 5 不必具備高深數學基礎的人就能輕易瞭解程式。
- 6 由於 COBOL 程式的簡易性，即使是非原編寫該程式的其他程式設計師亦能作有效而通盤性的測試。
- 7 由於 COBOL 程式本身就是一份敘述詳盡的英文語句，因此程式的作業說明易於編寫。
- 8 COBOL 程式的修改非常容易。

程式設計人員在學習 COBOL 程式語言時，必須具備某些重要的概念，而藉着以下三個問題可以有所啓示。

1. 電腦作業化需要些什麼條件？
2. 程式設計師在編寫 COBOL 程式時應經過那些步驟？
3. 程式編寫完成後應如何處理，才能使電腦正式應用此程式求得所要的結果？

1 - 2 練 習 題

COBOL 程式語言是專為_____或_____業務設計的一種易學易讀的高位準程式語言。

答：商業、與商業有關的

在不同廠牌電子計算機中_____ COBOL 程式語言，僅有少量的差異。

答：可以使用

3. COBOL 程式語言乃是使用一般 _____ 的術語編寫而成。

答：商業性

1.3 電腦作業化需要些什麼條件：

在目前這個自動化的世界裡，每當人們遭遇解析問題的時候，總是會求助於電腦的幫忙，然而殊不知有些問題適合於電腦作業化，有些卻不值得導入電腦作業，這都是由於電腦作業化有下列之特性：

1. 可以定義的 (Definable)
2. 可以重複的 (Repetitive)
3. 儲存或運算的空間 (Volume of storage or calculations)
4. 費用的衡量 (Cost-justifiable)

1-3-1 可以定義的 (Definable)

一個問題可經由一連串邏輯和運算步驟的清晰解說求得結果，即能適合導入電腦作業化，然而誠如事物的應用有所限制，電腦在運算的應用方面也是有其極限存在的。

1-3-2 可以重複的 (Repetitive)

如果一件事物的處理是週而復始的，例如某公司週薪的發放即是一個典型的例子，公司裡每個員工都需經過相同薪津的計算，僅僅是員工代號數目字的不同而已，就公司整體而言，一年要重複52次的同樣薪津作業，因此如果要以電腦來處理，資料的準備，程式的編寫，作業說明的建立所花費的成本將是分攤在一年內的使用次數上，除此之外，以人工來處理重複枯燥的計算必然會導致厭煩，而電腦機械性的操作是絕不會有此顧慮的。

1-3-3 儲存或運算的空間 (Volume of storage or calculations)

適合於電腦作業通常都需要大量的輸出入資料，如此龐大的資料需要儲存的空間，高效率的傳送設備方能作有效的運算處理，目前由於電腦快速的技術成長，可藉著作業手冊說明，打卡機，以及其他「統賦予」的設備，能夠有效的解決問題。

1-3-4 費用的衡量 (Cost justifiable)

作業資料的準備，程式的編寫，電腦執行程式指令，在在都有費用的產生，作業電腦化最終的結果，是輸入資料建立，輸出資料的獲得，時間及費用的節省，然而作業電腦化並不經常是節省時間和費用，許多人爲的工作甚至比作業電腦化來得更有效率，一旦發覺某項事物適合於電腦作業，則一切與電腦作業有關的準備就次第展開。

1 - 3 練習題

1. 電腦（並非經常是 / 經常是）能解決問題節省費用最有效的方法。

答：並非經常是

2. 適合於作業電腦化的特徵是 _____，_____ 儲存或運算的 _____，以及 _____。

答：可以定義的，可以重複的，空間，費用的衡量

3. 電腦作業化通常都需要大量的輸出入資料，要龐大的儲存空間，高效率的傳送設備方能做有效的運算處理，因此須考慮 _____。

答：費用的衡量

4. 一個問題可經由一連串邏輯和運算步驟予以清晰的說明，求得結果，則可說此問題是 _____。

答：可以定義的

5. 如果一件事情是重複不斷的發生，則可說是 _____。

答：可以重複的

1.4 由問題分析到作業說明的記載

程式設計師在作業導入電腦前，必須準備及考慮許多細節，諸如需要那些輸入資料，如何決定輸入資料格式，需要何種輸出媒體，產生那些輸出結果等是以必須要對電腦本身設備的特性，能力有所認識，才能把這些問題作一番通盤性的分析，詳細瞭解每一個步驟的細節，大體而言，概略可分爲 4 個項目：

1. 問題分析 (Problem analysis)
2. 作業系統流程 (Flowcharting)
3. 程式編寫，除錯及執行 (Coding debugging, and executing the problem)
4. 作業說明記載 (Documentation)

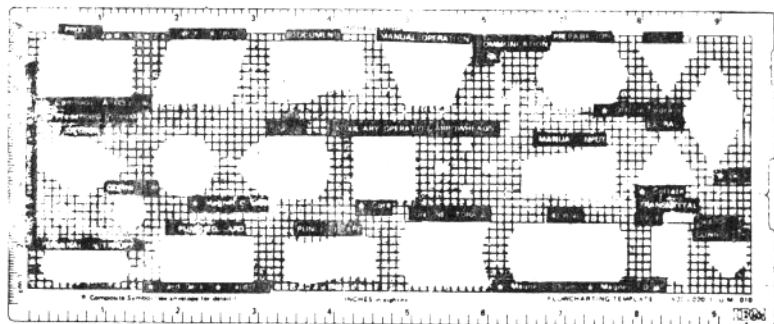
1-4-1 問題分析 (Problem analysis)

在作業導入電腦前，程式設計師必須確定問題，首先和使用單位研究是否作業電腦化是最適合、最具效益的方案，如果雙方都認為有此需要，程式設計師則進一步分析原始資料來源、規劃整體計劃大綱，決定輸出資料格式性質，由於使用單位對於本身作業認識最為透徹，能夠隨時密切協調合作，就能避免日後許多不必要的計劃修訂，甚至延誤目標的達成，問題分析完成後緊接著開始系統流程的設計。

1-4-2 作業系統流程 (Flow charting)

正如建築師設計建築物的工作藍圖，所謂系統流程 (Flowchart) 乃是利用圖形來代表一切資料計算和邏輯的操作，逐步說明如何運用輸入資料而得出需要的結果，建築師必須依照工作藍圖施工，而程式設計師則必須遵照系統流程來設計程式，製圖必須運用工具，圖形 1-1, 1-2 就是介紹美國標準協會制定的系統流程圖符號，定義及製作工具。

【圖形 1-2 系統流程圖設計工具】



程式設計師運用系統流程圖的益處如下：

- 1 提供電腦作業的藍圖指引。
- 2 提供在程式編寫、測試、邏輯推理方面的參考。
- 3 提供一種通則，使得程式設計師，系統設計分析師，及使用單位經辦人員彼此能夠易於意見溝通。
- 4 提供一種清晰的程式邏輯圖形，使得日後程式設計師易於維護程式。
- 5 是作業說明記載重要的一部份。

1-4-3 程式編寫、除錯及執行 (Coding , Debugging , and Executing The problem)

俟完成問題分析，設定系統流程圖後，程式設計師就要據此來安排一連串有意義及次序性的指令，亦即所謂的 COBOL 指令，程式編寫完成後，就要開始測試及除錯的工作，由於除錯狀況相複雜，無法具體說明除錯的詳細步驟，不過可以說除錯的工作就和偵探竭盡所能運用一切方法來推理解析所有可疑的線索是相同的，因此程式設計師必須花費工夫去準備測試資料，務必把一切隱藏的錯誤找出來，然而或許有些狀況無法獲悉致使有些錯誤無法偵測出，可能好幾天，好幾個月，甚至幾年後才會偶然的被發覺。

測試完成後，再拿實際的資料再做一番測試，程式設計師和使用單位經辦人員就得出的結果共同分析其正確性。

如果測試的是一個龐大而複雜的程式時，最好的方法是分段處理，分段測試，那麼測試工作就會變得較為單純，容易發掘錯誤，或許某段偵測出來的錯誤導致其它段落減少互為因果的其它錯誤發生，分段測試完成後再做次

整體性的除錯，俟測試完成後即可從事最後的作業說明記載。

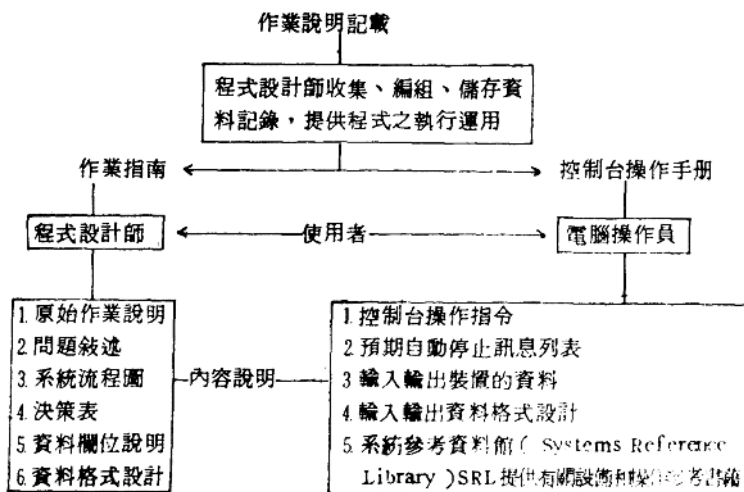
1-4-4 作業說明記載 (Documentation)

由於實際作業裡資料的情況變化多端，程式設計師不能保證程式計劃涵蓋了所有狀況，然而程式經過測試，建立作業說明記載後，則可謂是可操作的程式了。

作業說明由作業指南，即輸入輸出資料格式說明，程式系統流程圖，問題與解決的完整過程構成，所謂程式的作業手冊，如圖 1-3 所示，完整的作業說明對於凡和該程式有關的人員都是一種極為有用的工具，例如可以幫助低層管理人員對於問題及解決方法之認識，協助會計人員執行必要的帳目查核，並且提供日後更改程式的參考構架，在在都顯示完整作業 明的重要性了。

除此之外，儘管經常更改維護程式硬常常變動作業說明細節不過祇要有完整的作業說明，維護及修改程式遠比重新編寫程式要來得容易，同時費用也較低。

圖 1-3. 程式作業說明



1 - 4 練習題

1 程式設計師設計工作區分為_____，_____，_____和_____，和_____。

答：問題分析，作業系統流程，程式編寫除錯、執行，作業說明記載

2 在_____階段程式設計師首先和使用單位經辦人員研究是否作業電腦化是最適合、最具效益的方策。

答：問題分析

3 一旦決定是項作業適合於電腦處理，程式設計師則開始分析_____，規劃整體計劃大綱，決定輸出資料格式性質。

答：原始資料來源

4 _____能夠對電腦作業處理提供計算和邏輯操作的圖形構架。

答：程式系統流程圖

5 當程式編寫完成，即開始_____和_____的工作。

答：測試，除錯

6 在程式中設法找出並改正錯誤的方法，稱之為_____。

答：除錯

7 如果測試的是一個龐大而複雜的程式，最好的方法是_____，_____。

答：分段處理，分段測試

8 程式在經過完整的_____及清晰的_____後便可開始正式作業了。

答：測試，作業說明記載

1.5 COBOL 程式編寫完成後的處理

俟程式系統流程建立後，即可在編寫紙 (Coding sheets) 上編寫程式，隨即製卡，此成疊卡片稱為原始程式，如圖 1-4 篇看輸入媒體裝置傳達到電腦，程式設計師在接到打好成疊的原始程式卡片，於是在卡片前後放置指示電腦作業的必要控制卡，稱之為“作業控制語言” (Job Control Language，簡稱 JCL)，可參閱附錄 C 有詳細說明，經過 COBOL 編譯程式 (Compiler) 編譯成電腦可以執行的指令，即所謂的機器語言 (Machine Language)，此種過程稱之謂程式編譯 (Compilation)。