PROBLEMS WITH COMPUTER SOLUTIONS USING STRUCTURED COBOL

DOUGLAS S. TUNG
LENA L. SHAM

PROBLEMS WITH COMPUTER SOLUTIONS USING STRUCTURED COBOL

DOUGLAS S. TUNG
LENA L. SHAM

The Chinese University Press
Hong Kong

International Standard Book Number: 962-201-288-4

The Chinese University Press
The Chinese University of Hong Kong
SHATIN, N.T., HONG KONG

Printing by Ngai Kwong Printing Co., Ltd.

Preface

This book is written for those who need to solve administrative data processing problems by computers using structured COBOL. It is written in connection with the text 'Elementary COBOL: A Structured Programming Approach' published by The Chinese University Press. Structured Programming is a discipline approach to program design. Strict adherence to these rules produces well-designed, understandable programs that are easy to maintain and modify. It is felt that the techniques of structured programming should be taught when programming is first taught, because students should pick up good programming habits from the beginning.

The book consists of fourteen chapters. Most of the chapters are organized around a set of solved exercises and programming problems. Chapters 1, 2, 6, 7 and 8 contain only exercises. Topics covered in each chapter are mentioned in the table of contents. In each chapter, solutions to the exercises and programming problems are given after the questions. The readers are urged to work each of these questions carefully and thoroughly before they turn to the solutions for checking. We feel that this book is particularly useful as a supplement to any standard text book in structured COBOL programming.

The programs in Chapters 2 through 10 are written in WATBOL-11 which is a fast in-core compiler from the University of Waterloo. The programs are compiled and executed on the RSTS system of PDP 11/70. For programs involving tape and disk files in Chapters 11 through 14, they are OS/VS1 COBOL programs using the OS/VS1 operating system of the IBM 3031. Readers should read the implementor manual to be familiar with the features of the compiler used in running their programs.

We wish to express our gratitude to many people for their help in the preparation of this book: particularly to Professor S. C. Loh, Dr. H. S. Hung, Dr. K. W. Ng, Dr. Y. S. Moon, Dr. Y. K. Chan, Dr. Y. W. Ng, Mr. L. Kong and Mr. Henry Lam of the Department of Computer Science, The Chinese University of Hong Kong, for their encouragement and inspiration; to Mrs. L.K.H. Chim, Senior Lecturer of the Department of Computer Studies, Hong Kong Polytechnic and Miss B. Chan of the Department of Computer Studies, University of Hong Kong for their interest. We are also indebted to the following people who have contributed, either directly or indirectly, prior to and during the development of this book. Among these are Mr. W. H. Lo, Miss Irene W. F. Chan, Mr. W. K. Chong, Mr. Joseph Tu, Mr. Y. K. Fong and Miss C. S. Hu. Thanks are due also to Mrs. Cynthia Kwok and Mrs. M. O. Law for typing the original manuscript.

We must emphasize we are entirely responsible for any errors of commission and omission that may exist in the book.

Finally we would like to thank the United College for permission to reproduce the appendix of an article with the title 'The First Programming Language' by Douglas S. Tung on the comparison of BASIC, FORTRAN, COBOL and PASCAL, published in United Bulletin 36, August 1980, as the appendix of this book.

Douglas S. Tung
Lena L. Sham

August, 1982

Acknowledgment

The following information is reproduced from American National Standard Programming Language COBOL, published by the American National Standard Institute, Inc.

"Any organization interested in reproducing the COBOL report and specifications in whole or in part, using ideas taken from this report as the basis for an instruction manual or for any other purpose is free to do so. However, all such organizations are requested to reproduce this section as part of the introduction to the document. Those using a short passage, as in a book review, are requested to mention 'COBOL' in acknowledgment of the source, but need not quote this entire section.

"COBOL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations.

"No warrantly, expressed or implied, is made by any contributor or by the COBOL Committee as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection therewith.

"Procedures have been established for the maintenance of COBOL. Inquiries concerning the procedures for proposing changes should be directed to the Executive Committee of the Conference on Data Systems Languages.

"The authors and copyright holders of the copyrighted material used herein

> FLOW-MATIC (Trademark of Sperry Rand Corporation), Programming for the Univac (R) I and II, Data Automation Systems copyrighted 1958, 1959, by Sperry Rand Corporation: IBM Commercial Translator Form No. F 28-8013, copyrighted 1959 by IBM; FACT, DSI 27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell

have specifically authorized the use of this material in whole or in part, in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications."

Table of Contents

Chapter 1 Background Concepts

EXERCISES:

1. Obtain a description of the computer that is available to you to compile and execute your COBOL program. List the various storage and I/O units. Does the description contain any other peripherals?
2. What version of COBOL is available to you to compile and execute your program? Find out as much information as possible on the COBOL compiler to be used for running your programs from the consultants in the computer centre.
3. Why has COBOL emerged as the leading language in the data processing community?
4. Is COBOL as efficient as a program written in the machine language for a particular computer? Explain.
5. What groups were represented at the early meetings held to standardize COBOL? What agency has published the specifications for COBOL?

SOLUTIONS TO EXERCISES:

1. One of the computer systems used in this book is a PDP-11/70 which is the most powerful computer in the PDP-11 family, providing multiuser and multitasking environment. Installed with the CPU (Central Processing Unit) are 512 Kb MOS (Metal-oxide semiconductor) memory and 2 Kb of high-speed Cache memory.

   Attached to the processor are two RH70 magnetic disk controllers, one magnetic tape controller, one CR-11B card reader and two DH-11 serial line multiplexers.

   The line printer is an IBM model 1403 feature 2 capable of printing up to 750 lines per minute with 132 print positions. A LSI-11 micro processor run as an emulater to interface the IBM printer control with the Unibus on the PDP.

   The card reader is a CR-11B model capable of up to 600 cards per minute operation.

   The disk module is a RM-03 disk pack subsystem with the following characteristics:

Storage capacity	67 Mb
Data transfer rate	1.2 Mb/sec
Average access-motion time	25 millisec
Average rotational delay	8.3 sec

The magnetic tape unit is a TE-16 nine-track model with a transfer rate of above 72,000 bytes per sec, a capacity of 40 Mb, recording density of 800/1600 bytes per sec (program selectable) and a read/write speed of 45 inches per sec.

Attached to the DH-11 terminal control units are 30 Hazeltine 1500 and 1510 CRT's and two Dec-writers.

The operating system running on the PDP-11/70 is RSTS/E (Resource Sharing Time System/Extend) providing simultaneous interactive program development and testing, time-sharing and batch processing. A variety of programming languages such as BASIC PLUS, WATFOR, WATBOL and MACRO-11 are provided for student use.

2. WATBOL-11 is a fast in-core compiler from the University of Waterloo, operating on the PDP-11. It is a special implementation of the COBOL language which has been designed for use in an educational environment. It can only be used to create batch applications.

Language elements :

WATBOL-11 conforms in language element, representation, symbology, and coding format to ANSI-74 COBOL, including:

1. A full high-level Nucleus module, providing all language elements necessary for internal processing.

2. A full high-level table handling module for defining and manipulating tabular data. The SEARCH verb will be supported in future versions.

3. Full high-level sequential and relative I/O modules for defining and accessing sequential and relative files.

4. A full low-level library function for copying predefined COBOL text into the source program.

```
$COPY file-name
```

5. Conditional variables.

6. Nested conditionals.

7. Extensions:

   (a) Separators, period, comma and semicolon may optionally be preceded by a space.

   (b) Non-numeric literals may be enclosed in either single quotes(') or double quotes("), but not a combination of both. Quotes can be placed within a non-numeric literal by enclosing the literal in the other type of quote symbol. For example "ABC'DE" OR 'ABC"DE'.

   (c) A symbolic name may be used for file specification in the ASSIGN clause of the FILE-CONTROL paragraph. The external file name is specified with the $FILE control statement. (Refer to WATBOL-11 User's Guide.)

   (d) More than the standard three level of subscriptions are allowed.

   (e) Level 77 and level 01 elementary items may contain an OCCURS clause.

   (f) Expressions of the form 'subscript +/- literals' are permitted; subscript and index names may be mixed in a subscript expression.

8. Restrictions:

   (a) No RENAMES in the Data Division.

   (b) No SPECIAL-NAMES in the Environment Division.

(c) No subroutines.

(d) The following are not allowed in the Procedure Division :

(1) Declarative section
(2) ALTER
(3) switch-status condition
(4) PERFORM-VARYING statement with AFTER
(5) ACCEPT-FROM, DISPLAY-UPON
(6) OPEN and CLOSE with options REEL, UNTIL, NO REWIND, LOCK
(7) WRITE with END-OF-PAGE option

3. COBOL has emerged as the leading language in the data processing community because

- COBOL is the only language translator supported by the U.S Federal Government;
- COBOL has been continuously standardized by repeated CODASYL(Conference on Data Systems Languages) meetings;
- COBOL has a self-documentary feature with the English language statements being easily understood by nonprogramming personnel.

4. COBOL is not as efficient as program written in machine language, because it must spend time to translate COBOL into machine language, which is the only language that the computer can understand.

5. They are groups of computer users, manufacturers and universities representatives.
CODASYL has published the specifications for COBOL.

Chapter 2 Introduction to COBOL Programming

EXERCISES:

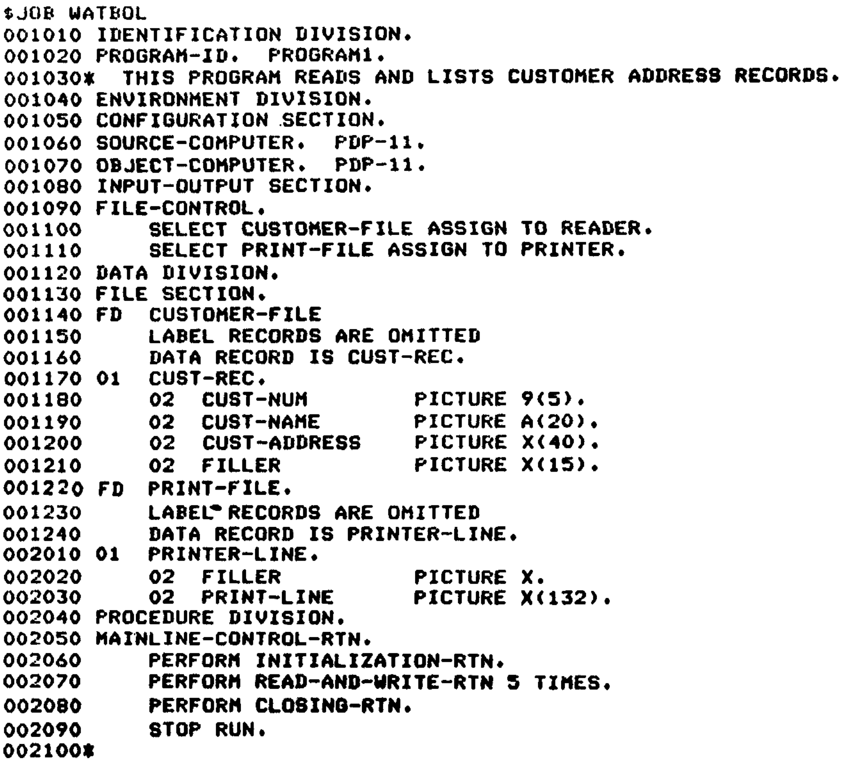
1. Circle the legitimate data-names in COBOL (exclude paragraph name):

```
EMPLOYEE NUM
23B5
23-B5
AMOUNT-
23
END-CF-YEAR-BALANCE-CUE-ON-ACCOUNT
UNTIL
PROCECURE-DIVISION
TAX-%
BALANCE
```

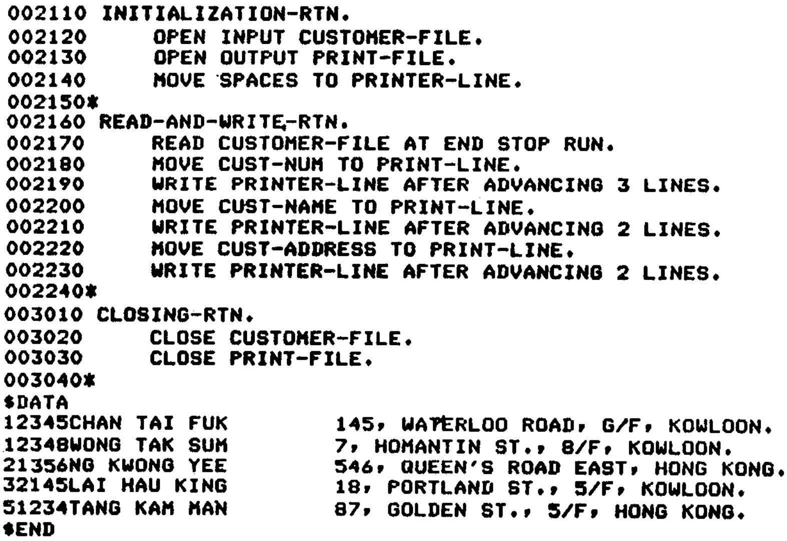
2. In the COBOL Coding Sheet, which of the following can/must start in Areas A and/or B?

Division	Sentence
Section	Statement
Paragraph	

3. The address label program shown as follows is to be prepared.



```
$JOB WATBOL
001010 IDENTIFICATION DIVISION.
001020 PROGRAM-ID.  PROGRAM1.
001030*  THIS PROGRAM READS AND LISTS CUSTOMER ADDRESS RECORDS.
001040 ENVIRONMENT DIVISION.
001050 CONFIGURATION SECTION.
001060 SOURCE-COMPUTER.  PDP-11.
001070 OBJECT-COMPUTER.  PDP-11.
001080 INPUT-OUTPUT SECTION.
001090 FILE-CONTROL.
001100     SELECT CUSTOMER-FILE ASSIGN TO READER.
001110     SELECT PRINT-FILE ASSIGN TO PRINTER.
001120 DATA DIVISION.
001130 FILE SECTION.
001140 FD  CUSTOMER-FILE
001150     LABEL RECORDS ARE OMITTED
001160     DATA RECORD IS CUST-REC.
001170 01  CUST-REC.
001180     02  CUST-NUM         PICTURE 9(5).
001190     02  CUST-NAME        PICTURE A(20).
001200     02  CUST-ADDRESS     PICTURE X(40).
001210     02  FILLER           PICTURE X(15).
001220 FD  PRINT-FILE.
001230     LABEL RECORDS ARE OMITTED
001240     DATA RECORD IS PRINTER-LINE.
002010 01  PRINTER-LINE.
002020     02  FILLER           PICTURE X.
002030     02  PRINT-LINE       PICTURE X(132).
002040 PROCEDURE DIVISION.
002050 MAINLINE-CONTROL-RTN.
002060     PERFORM INITIALIZATION-RTN.
002070     PERFORM READ-AND-WRITE-RTN 5 TIMES.
002080     PERFORM CLOSING-RTN.
002090     STOP RUN.
002100*
```



```
002110 INITIALIZATION-RTN.
002120     OPEN INPUT CUSTOMER-FILE.
002130     OPEN OUTPUT PRINT-FILE.
002140     MOVE SPACES TO PRINTER-LINE.
002150*
002160 READ-AND-WRITE-RTN.
002170     READ CUSTOMER-FILE AT END STOP RUN.
002180     MOVE CUST-NUM TO PRINT-LINE.
002190     WRITE PRINTER-LINE AFTER ADVANCING 3 LINES.
002200     MOVE CUST-NAME TO PRINT-LINE.
002210     WRITE PRINTER-LINE AFTER ADVANCING 2 LINES.
002220     MOVE CUST-ADDRESS TO PRINT-LINE.
002230     WRITE PRINTER-LINE AFTER ADVANCING 2 LINES.
002240*
003010 CLOSING-RTN.
003020     CLOSE CUSTOMER-FILE.
003030     CLOSE PRINT-FILE.
003040*
$DATA
12345CHAN TAI FUK          145, WATERLOO ROAD, G/F, KOWLOON.
12348WONG TAK SUM          7, HOMANTIN ST., 8/F, KOWLOON.
21356NG KWONG YEE          546, QUEEN'S ROAD EAST, HONG KONG.
32145LAI HAU KING          18, PORTLAND ST., 5/F, KOWLOON.
51234TANG KAM MAN          87, GOLDEN ST., 5/F, HONG KONG.
$END
```

Fig. 2.1 Address labels program .

As most of the procedure is highly system-dependent, students should obtain a description of the correct job control cards, correct names for the SOURCE-COMPUTER and OBJECT-COMPUTER, and implementor-names assigned to card input and printer output from the installation.

In addition, the module names should be changed as follows:

Old Module-name	New Module-name
MAINLINE-CONTROL-ROUTINE	MAINLINE
INITIALIZATION	HOUSEKEEPING
READ-AND-PRINT	READ-PRINT
CLOSING	CLOSE-FILES

4. As a challenge, make changes in the file-names in addition to changing the module names.

Old File-name	New File-name
CUSTOMER-FILE	INPUT-FILE
PRINT-FILE	REPORT-FILE

5. What columns on a COBOL coding sheet indicate a sequencing of instructions?
6. What are the functions of columns 73-80 of a COBOL coding sheet?

SOLUTIONS TO EXERCISES:

```
1.         EMPLOYEE NUM
         o 23B5
         o 23-B5
           AMOUNT-
           23
           END-OF-YEAR-BALANCE-DUE-ON-ACCOUNT
           UNTIL
         o PROCEDURE-DIVISION
           TAX-%
         o BALANCE

2.       Division\
         Section --    can start in margin A or A-area
         Paragraph/

         Sentence --   can start in Margin B or B-area
         Statement/


3. Change should be made in the PROCEDURE DIVISION
   as listed in the COBOL coding form.

         SOURCE-COMPUTER.  PDP-11-70.
         OBJECT-COMPUTER.  PDP-11-70.
         INPUT-OUTPUT SECTION.
         FILE-CONTROL.
              SELECT CUSTOMER-FILE ASSIGN TO READER.
              SELECT PRINT-FILE ASSIGN TO PRINTER.
       *
           .
           .
           .
       * PROCEDURE DIVISION.
         MAINLINE.
             PERFORM HOUSEKEEPING-RTN.
             PERFORM READ-PRINT-RTN 5 TIMES.
             PERFORM CLOSE-FILES-RTN.
             STOP RUN.

      * HOUSEKEEPING-RTN.
             OPEN INPUT CUSTOMER-FILE.
             OPEN OUTPUT PRINT-FILE.
             MOVES SPACES TO PRINTER-LINE.

      * READ-PRINT-RTN.
             READ CUSTOMER-FILE AT END STOP RUN.
             MOVE CUST-NUM TO PRINT-LINE.
             WRITE PRINTER-LINE AFTER ADVANCING 3 LINES.
             MOVE CUST-NAME TO PRINT-LINE.
             WRTIE PRINTER-LINE AFTER ADVANCING 2 LINES.
             MOVE CUST-ADDRESS TO PRINT-LINE.
             WRITE PRINTER-LINE AFTER ADVANCING 2 LINE.
      *
```

```
     CLOSE-FILES-RTN.
          CLOSE CUSTOMER-FILE.
          CLOSE PRINT-FILE.
   *
```

```
4. 00101 IDENTIFICATION DIVISION.
       .
       .
       .
       09 FILE-CONTROL.
       10     SELECT INPUT-FILE ASSIGN TO SUS005-UR-2501-S.
       11     SELECT REPORT-FILE ASSIGN TO SYS006-UR-1405-S.
       12 DATA DIVISION.
       13 FILE SECTION.
       14 FD  INPUT-FILE
        .     .
        .     .
        .     .
       22 FD  REPORT-FILE
        .     .
        .     .
        .     .
   00204 PROCEDURE DIVISION
       05 MAINLINE-RTN.
       06     PERFORM HOUSEKEEPING-RTN.
       07     READ-PRINT-RTN 5 TIMES.
       08     PERFORM CLOSE-FILES-RTN.
       09     STOP RUN.
       10 *
       11 HOUSEKEEPING-RTN.
       12     OPEN INPUT INPUT-FILE.
       13     OPEN OUTPUT REPORT-FILE.
        .        .
        .        .
        .        .
       16 READ-PRINT-RTN.
       17     READ INPUT-FILE AT END STOP RUN.
        .        .
        .        .
        .        .
   00301 CLOSE-FILES-RTN.
       02     CLOSE INPUT-FILE.
       03     CLOSE REPORT-FILE.
```

5. Columns 1 to 6.

6. It serves as an identification.

Chapter 3 Basic Elements of COBOL

EXERCISES:

1. Circle the legitimate numeric literals:

+3,241
−2/3
0.000001
+12.0
−23459.

439.09
0.3E3
56.2−
+.3451
12345678911234567890

2. Circle the legitimate non-numeric literals:

'BRANDY OR COGNAC'
'2.3E3'
'$**1,234.56'

ABC'
' '
'PETER'S BOOK'

3. In the following list, place an NL before those expressions that could serve as a numeric literal in a COBOL program, an FC for figurative constants an NON-L for non-numeric literals, and an X for expressions not exemplifying any of these language elements.

________ GROSS-pay
________ '12%'
________ 123.4
________ $100.00
________ ZEROS

________ 'ENVIRONMENT DIVISION'
________ 'SPACES'
________ SPACES
________ '123.4'
________ +72.

4. a. Write the Record Description entries for the following:

	TRANSACTION RECORD						
	INVOICE NUMBER	PRODUCT DESCRIPTION			LOCATION		
		NUMBER		NAME	WARE-HOUSE NAME	ADDRESS	
		SIZE	MODEL			STREET	CITY
Card Columns	1–5	6–8	9–18	19–38	39–50	51–68	69–78