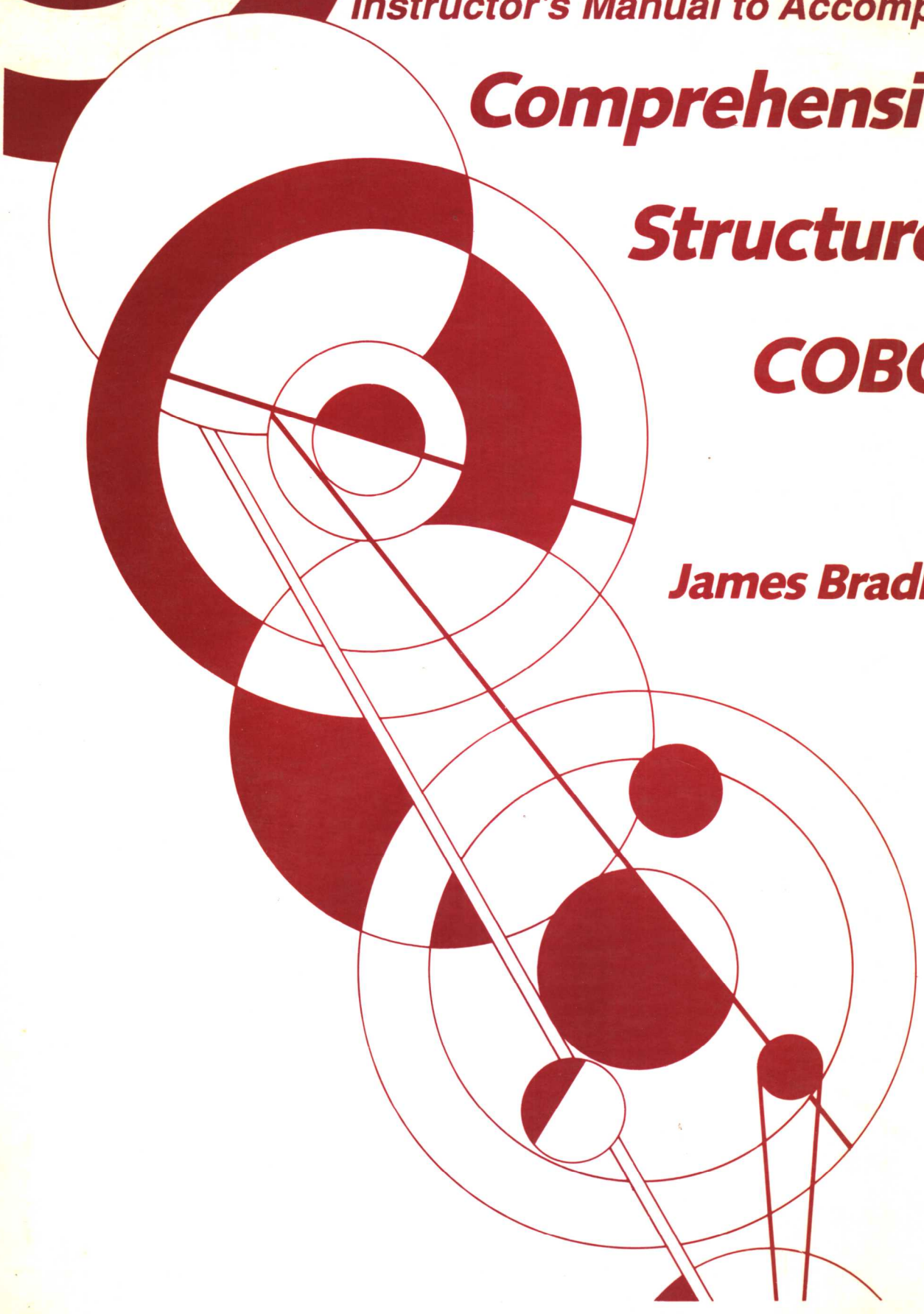
Instructor's Manual to Accompany

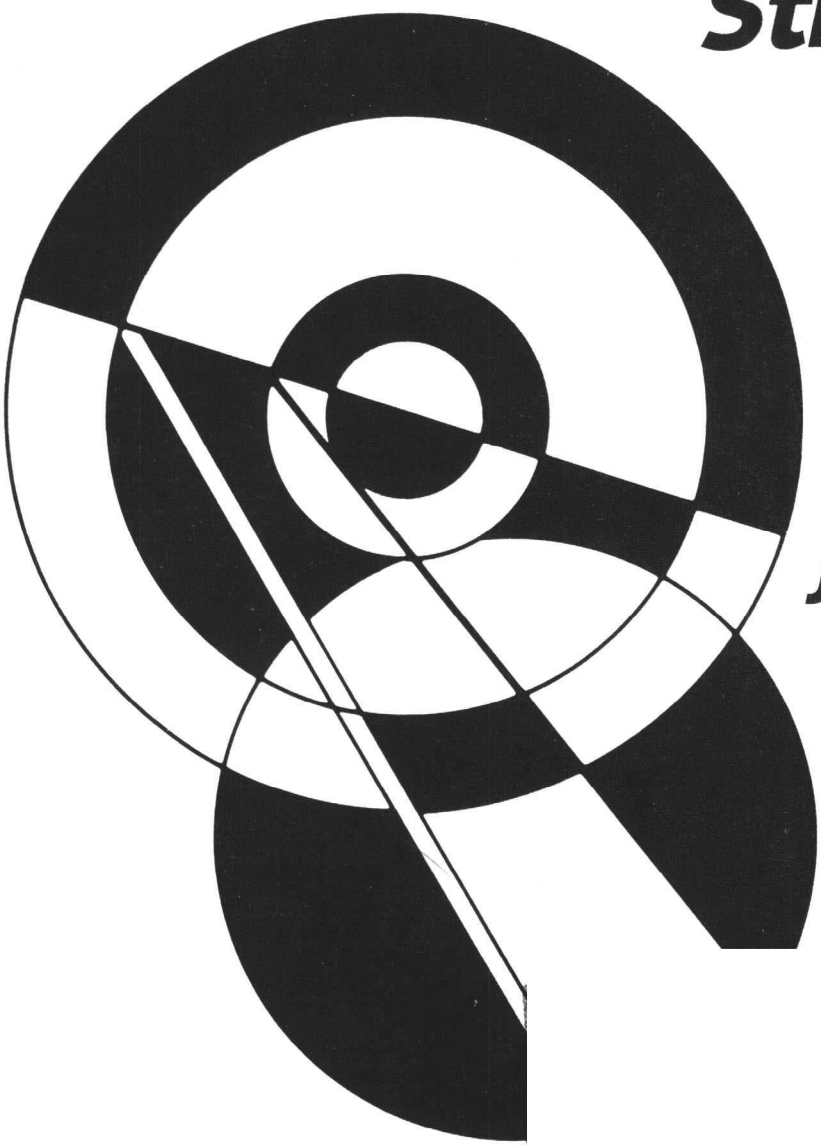
Comprehensive Structured COBOL



James Bradley

Instructor's Manual to Accompany

Comprehensive Structured COBOL



James Bradley

University of Calgary

Mitchell McGRAW-HILL

New York St. Louis San Francisco Auckland Bogotá Caracas
Hamburg Lisbon London Madrid Mexico Milan Montreal
New Delhi Oklahoma City Paris San Juan São Paulo
Singapore Sydney Tokyo Toronto Watsonville

Mitchell **McGraw-Hill**
55 Penny Lane
Watsonville, California 95076

Instructor's Manual to accompany **Comprehensive Structured COBOL**

2 3 4 5 6 7 8 9 0 HAN HAN 9 5 4 3 2

ISBN 0-07-007079-2

Information for Users of the Data Disk

1. File names relate to the chapter first and assignment second. For example, file c10e2 is for assignment 2 in Chapter 10.

2. On occasion, the length of the records cannot be deduced from the data in the text, because the last field is alphanumeric and varies in length. In such cases, the records are padded with trailing blanks, so they are a fixed length, and match the record length assumed in the solution program given in the Instructor's Manual. In addition, in such cases a note (or document) file is included to document the lengths of the records. For example, the file notec10 gives record lengths for files in Chapter 10.

3. Every record ends in the conventional ASCII end-of-line character, whether padded with blanks or not.

CONTENTS

CHAPTER 1

ELEMENTARY COBOL DATA PROCESSING CONCEPTS

For students with no prior programming experience

Teaching Guide

The most important thing is for students to understand the difference between input data, output data, and the collection of processing steps required to generate the output data from the input. Use the analogy of a manufacturing plant to drive the point home (beginning with raw materials, going through the steps involved in processing, and ending with the production of finished goods).

Be careful to remove the abstraction from data processing. For example, show that input data is recorded on disk or tape, character by character, in a special magnetic code. Show some ASCII (or EBCDIC codes). Explain how instructions are stored in memory and are executed in sequence by a processor. Explain that an instruction can move data from disk or tape to memory, can alter or carry out arithmetic on data in memory, and can transfer data from memory to disk or tape. But underscore that no instruction can directly alter or perform arithmetic on data on disk or tape. The data must first be read into memory, processed, and then written out.

Explain that instructions and data being processed are stored in distinct parts of memory. Data being processed is held in locations in a part of memory called *working storage*. Explain that each location in working storage has a distinct address, so that an instruction can refer to the contents of that address. Explain also, that in COBOL, arbitrary names, called *identifiers*, are used in programs instead of addresses, and that the COBOL compiler substitutes the same address in every instruction involving a given identifier.

Explain the function of a compiler: how it converts instructions in COBOL into machine language instructions. Explain that machine language instruction is coded in binary format in memory, by electrical means, and contains a binary code for the actual operation, such as MOVE, ADD, SUBTRACT, together with binary address codes for the addresses of the locations for which the contents are involved in the operation.

Next you might use the material that follows for students with prior programming experience. Then explain how the logic of a program can be illustrated by a flowchart. Finally, make the students trace through some simple programs using the desk checking technique, the quickest and easiest way to understand just what goes on as a program executes and to help students gain confidence.

Topics for Classroom Discussion

1. How data gets from documents to disk and tape files prior to computer processing, and how much time this takes.
2. The possible ways in which data could be coded on magnetic tape.
3. The contrast between main memory in a computer and disk and tape "memory."
4. The consequences of the fact that an instruction in memory operates on the contents of named memory locations.
5. Why you must always use the same name in a program for a specific memory location.
6. The analogy between manufacturing and data processing.
7. How flowcharts can be used in manufacturing, data processing and any other operation, such as changing a tire.
8. Why learning to do desk checking takes the mystery out of computing.
9. What different kinds of errors are to be expected in a program and where they are detected.

For students with prior programming experience

Teaching Guide

Explain how COBOL is normally used for processing relatively large input files, formatted into records, to generate formatted output files in the form of reports or disk/tape files or both. Consequently, lengthy specifications of the variables or identifiers corresponding to the fields of the input and output files are needed.

Use the program in Figure 1.8 as an example of the processing concepts involved. Use it to show the four divisions of a COBOL program. Distinguish carefully between the part of memory holding the input (output) area or input (output) buffer and the part holding the working storage. Point out that identifiers corresponding to locations in the input (output) buffer are specified in the FILE SECTION of the DATA DIVISION, and identifiers corresponding to locations in working storage are specified in the WORKING-STORAGE SECTION of the DATA DIVISION. Show that the instructions are held in the final PROCEDURE DIVISION.

Underscore the importance of clear record layouts for input data printer spacing charts for output report data.

Remember to have the students enter and run a simple program initially to enable them to gain experience quickly with the editor, operating system, and COBOL compiler used in your course.

Topics for Classroom Discussion

1. Why the ENVIRONMENT and DATA divisions have such a formal structure in COBOL.
2. The distinction between input/output buffer identifiers and working storage identifiers.
3. Why input record layouts and printer spacing charts are so important with COBOL programs.

Solutions to COBOL Language Questions

7. a. 31 b. 6 c. 25
10. OPEN INPUT IN-FILE.
11. OPEN OUTPUT PRINTER-FILE.
13. MOVE 'NO' TO NPT-FOUND.
14. WRITE PR-RECORD-OUT FROM DETAIL-LINE.
15. 99,999.
16. a. 15 b. 5 c. 53 d. 280
19. a. MOVE A TO B.
    b. MOVE A1 TO B1
       MOVE A2 TO B2.
20. FILLER cannot be referenced in an instruction (or statement).

Solutions to Programming Assignments

Assignment 4.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. C1E4.

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.    SELECT SOURCEFILE-IN     ASSIGN TO UT-S-name-1.
                 SELECT REPORT-OUT        ASSIGN TO UR-S-name-2.

DATA DIVISION.
FILE SECTION.

FD  SOURCEFILE-IN                         LABEL RECORDS ARE STANDARD.
01  SF-RECORD-IN.
    05 SF-CUSTNAME-IN                     PIC  X(20).
    05 FILLER                             PIC  X(03).
```

```
    05 SF-SAVINGS-IN                    PIC  9(05).
    05 FILLER                           PIC  X(03).
    05 SF-CHECKING-IN                   PIC  9(05).

FD  REPORT-OUT                          LABEL RECORDS ARE OMITTED.
01  REPORT-RECORD-OUT                  PIC X(133).

WORKING-STORAGE SECTION.
01 WS-AREA.
    05 MORE-RECORDS                     PIC X(03)        VALUE 'YES'.
    05 WS-TOTAL                         PIC 9(06).

01  DETAIL-LINE-OUT.
    05 FILLER                           PIC X(01)        VALUE SPACES.
    05 DL-CUSTNAME-OUT                  PIC X(20).
    05 FILLER                           PIC X(05)        VALUE SPACES.
    05 DL-SAVINGS-OUT                   PIC 9(05).
    05 FILLER                           PIC X(08)        VALUE SPACES.
    05 DL-CHECKING-OUT                  PIC 9(05).
    05 FILLER                           PIC X(06)        VALUE SPACES.
    05 DL-TOTAL-OUT                     PIC 9(06).
    05 FILLER                           PIC X(77)        VALUE SPACES.

PROCEDURE DIVISION.
100-MAIN-MODULE.
    OPEN INPUT SOURCEFILE-IN
         OUTPUT REPORT-OUT.
    READ SOURCEFILE-IN
              AT END MOVE 'NO' TO MORE-RECORDS.
    PERFORM 200-PROCESS-RECORD
           UNTIL MORE-RECORDS = 'NO'.
    CLOSE SOURCEFILE-IN
          REPORT-OUT.
    STOP RUN.

200-PROCESS-RECORD.
    COMPUTE WS-TOTAL = SF-SAVINGS-IN + SF-CHECKING-IN.
    MOVE WS-TOTAL TO DL-TOTAL-OUT.
    MOVE    SF-CUSTNAME-IN    TO       DL-CUSTNAME-OUT.
    MOVE    SF-SAVINGS-IN     TO       DL-SAVINGS-OUT.
    MOVE    SF-CHECKING-IN    TO       DL-CHECKING-OUT.
    WRITE REPORT-RECORD-OUT FROM DETAIL-LINE-OUT
                     AFTER ADVANCING 1 LINE.
    READ SOURCEFILE-IN
                     AT END MOVE 'NO' TO MORE-RECORDS.
```

Examination Materials

True/False Questions

1. A compiler is used for compiling data prior to computer processing. ANSWER F
2. The instructions needed for processing a tape file of input data are always stored on the tape along with the data. ANSWER F
3. Both instructions and data can be stored in computer memory. ANSWER T
4. Both instructions and data can be stored permanently in computer memory. ANSWER F
5. Both instructions and data can be stored permanently on tape or disk. ANSWER T
6. The bit code used to store a character on disk is the same as that used to store the character in computer memory. ANSWER T
7. A COBOL program has three divisions. ANSWER F
8. The WORKING-STORAGE DIVISION is one of the divisions of a COBOL program. ANSWER F
9. A COBOL identifier names a location in memory capable of holding data. ANSWER T
10. The use of A and B margins is optional with COBOL programs. ANSWER F

Multiple-Choice Questions

Pick the correct statement.

1. Which of the following is true?
   a. Computer memory holds instructions but not data.
   b. Computer memory holds data but not instructions.
   c. Computer memory holds both instructions and data.
   d. Computer memory holds neither instructions nor data.
   ANSWER c
2. Which of the following is true?
   a. An identifier identifies a computer instruction.
   b. An identifier identifies a COBOL arithmetic instruction.
   c. An identifier identifies a location in memory.
   d. An identifier identifies a program.
   ANSWER c

3. Which of the following is true?
   a. A VALUE clause is used with an instruction.
   b. A VALUE clause is used with a paragraph.
   c. A VALUE clause is used to name an identifier.
   d. A VALUE clause is used to place an initial value in an identifier.

   ANSWER d
4. Which of the following is true?
   a. The PROCEDURE DIVISION contains procedures for defining identifiers.
   b. The PROCEDURE DIVISION contains instructions.
   c. The PROCEDURE DIVISION contains both instructions and identifier definitions.
   d. The PROCEDURE DIVISION contains at least one WORKING-STORAGE section.

   ANSWER b

Fill in the Blank

1. A COBOL compiler generates a machine language program.
2. An editor is used to input data and programs at a terminal.
3. An input area can hold a record read from an input file.
4. A character is stored in a byte of memory
5. A logical error may cause incorrect output.
6. When specifying the format of a report, a printer spacing chart is used to reduce errors.
7. A record is made up of fields.
8. A COBOL file is a collection of records.
9. WORKING STORAGE is the part of memory where identifiers are defined.
10. A COBOL paragraph contains COBOL instructions.

Debugging Questions

Find and correct the bugs, if any.

1.
```
OPEN SOURCEFILE-IN INPUT
     REPORT-OUT OUTPUT.
```

Should be:

```
    OPEN INPUT SOURCEFILE-IN
         OUTPUT REPORT-OUT.
```

2.

```
    OPEN INPUT SOURCEFILE-IN
         OUTPUT REPORT-OUT.
    READ SOURCEFILE-IN.
    PERFORM 200-PROCESS-RECORD
           UNTIL MORE-RECORDS = 'NO'.
```

Should have:

```
    READ SOURCEFILE-IN
              AT END MOVE 'NO' TO MORE-RECORDS.
```

3.

```
FILE-CONTROL.    SELECT CUSTOMER-IN       ASSIGN TO UT-S-name-1.
                 SELECT REPORT-OUT        ASSIGN TO UR-S-name-2.

DATA DIVISION.
FILE SECTION.

CUSTOMER-IN                  LABEL RECORDS ARE STANDARD.
01  CU-RECORD-IN.
    05 CU-CUSTNAME-IN PIC  X(20).
    05 FILLER                      PIC  X(03).
```

Should be:

```
FILE-CONTROL.   SELECT CUSTOMER-IN    ASSIGN TO UT-S-name-1.
...
FD  CUSTOMER-IN                  LABEL RECORDS ARE STANDARD.
```

4.

```
01 WS.
05 MORE-RECORDS PIC X(03) VALUE 'YES'.
05 TOTAL PIC 9(06).
```

Should be:

```
01 WS.
    05 MORE-RECORDS PIC X(03) VALUE 'YES'.
    05 TOTAL PIC 9(06).
```

Construction of COBOL Excerpts.

1. Print the following three values, from a record of an input file, plus the sum on a single line:
   15 21 45

```
FILE-CONTROL.    SELECT SOURCEFILE-IN     ASSIGN TO UT-S-name-1.
                 SELECT REPORT-OUT        ASSIGN TO UR-S-name-2.

DATA DIVISION.
FILE SECTION.

FD  SOURCEFILE-IN                  LABEL RECORDS ARE STANDARD.
01  SF-RECORD-IN.
    05 SF-INTEGER1-IN PIC  9(02).
    05 FILLER                      PIC  X(01).
    05 SF-INTEGER2-IN PIC  9(02).
    05 FILLER                      PIC  X(01).
    05 SF-INTEGER3-IN PIC  9(02).
    05 FILLER                      PIC  X(01).

FD  REPORT-OUT                     LABEL RECORDS ARE OMITTED.
01  REPORT-RECORD-OUT  PIC X(133).

WORKING-STORAGE SECTION.
01 WS-AREA.
    05 WS-TOTAL PIC 9(03).

01  DETAIL-LINE-OUT.
    05 FILLER                       PIC  X(01)   VALUE SPACES.
    05 DL-INTEGER1-OUT PIC  9(02).
    05 FILLER                      PIC  X(01)       VALUE SPACES.
    05 DL-INTEGER2-OUT PIC  9(02).
    05 FILLER                      PIC  X(01)       VALUE SPACES.
    05 DL-INTEGER3-OUT PIC  9(02).
    05 FILLER                      PIC  X(01)       VALUE SPACES.
    05 DL-TOTAL-OUT PIC  9(03).

PROCEDURE DIVISION.
100-MAIN-MODULE.
    OPEN INPUT SOURCEFILE-IN.
    READ SOURCEFILE-IN
            AT END MOVE 'NO' TO MORE-RECORDS.
    COMPUTE WS-TOTAL
      = SF-INTEGER1-IN + SF-INTEGER2-IN + SF-INTEGER-IN.
    MOVE SF-INTEGER1-IN TO DL-INTEGER1-OUT.
    MOVE SF-INTEGER2-IN TO DL-INTEGER2-OUT.
    MOVE SF-INTEGER3-IN TO DL-INTEGER3-OUT.
    MOVE WS-TOTAL TO DL-TOTAL-OUT.
    WRITE REPORT-RECORD-OUT FROM DETAIL-LINE-OUT
```

```
                    AFTER ADVANCING 1 LINE.
CLOSE SOURCEFILE-IN
      REPORT-OUT.
STOP RUN.
```

CHAPTER 2

THE IDENTIFICATION AND ENVIRONMENT DIVISIONS

Teaching Guide

The material on the IDENTIFICATION and ENVIRONMENT divisions is mainly reference material. But make sure you show students how to construct implementor names with ASSIGN clauses for your computer installation. If none of the material on implementor names suits your computer, you should give students a handout with the correct material.

The more interesting part of the chapter concerns how to construct headers for a single page report. Make sure students understand that all headers are printed via the output buffer record.

The other important topic introduced in this chaper has to do with edit items. Make sure that students understand the difference between an implicit (V) decimal point and an explicit (.) one.

Finally, give students a brief introduction to the arithmetic verbs.

Topics for Classroom Discussion

1. Why the file name as used in a program and the physical file name often have to be different.
2. The printing of a sequence of headers and detail lines via the same output area record.
3. The concept of an edited numeric identifier as opposed to an unedited identifier.
4. Why COBOL allows both COMPUTE as well as the ADD, SUBTRACT, and other computational verbs.

Solutions to COBOL Language Questions

1. Name is too long.
2. Should be PROGRAM-ID.
3. Second SELECT should begin in B-area.

4. Should be a period after CUST23.
5. Should be READ CUSTOMER-IN ...
6. 132 characters are printed, but with some computers 133 are transmitted to the printer.
7. $ 6.42.
8. PIC X(116).
9. 01547.
10. Replace MOVE and COMPUTE statements with:
    `COMPUTE Y = X * X.`

Solutions to Programming Assignments

Assignment 1.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. C2E1.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
               SELECT DISK-FILE-IN          ASSIGN TO UT-S-DISKIN.
               SELECT PRINTER-FILE-OUT      ASSIGN TO UR-S-SYSPRINT.

DATA DIVISION.
FILE SECTION.
FD  DISK-FILE-IN                        LABEL RECORDS ARE STANDARD.
01  DISK-RECORD-IN.
    05  DK-PARTNUMB-IN      PIC  X(05).
    05  FILLER              PIC  X(02).
    05  DK-UNIT-PRICE-IN    PIC  9(04).
    05  FILLER              PIC  X(02).
    05  DK-QUANTITY-IN      PIC  9(04).

FD  PRINTER-FILE-OUT                    LABEL RECORDS ARE OMITTED.
01  PRINTER-RECORD-OUT      PIC X(133).

WORKING-STORAGE SECTION.
01 WS-WORK-AREAS.
    05  MORE-RECORDS        PIC  X(03).
    05  COMMISSION          PIC  9(02)      VALUE 10.
    05  WS-COMPUTED-COST    PIC  9(09).

01  HEADER.
    05 FILLER                          PIC X(17)        VALUE SPACES.
    05 FILLER                          PIC X(15)
                VALUE 'SALES REPORT'.
    05 FILLER                          PIC X(101)       VALUE SPACES.
```