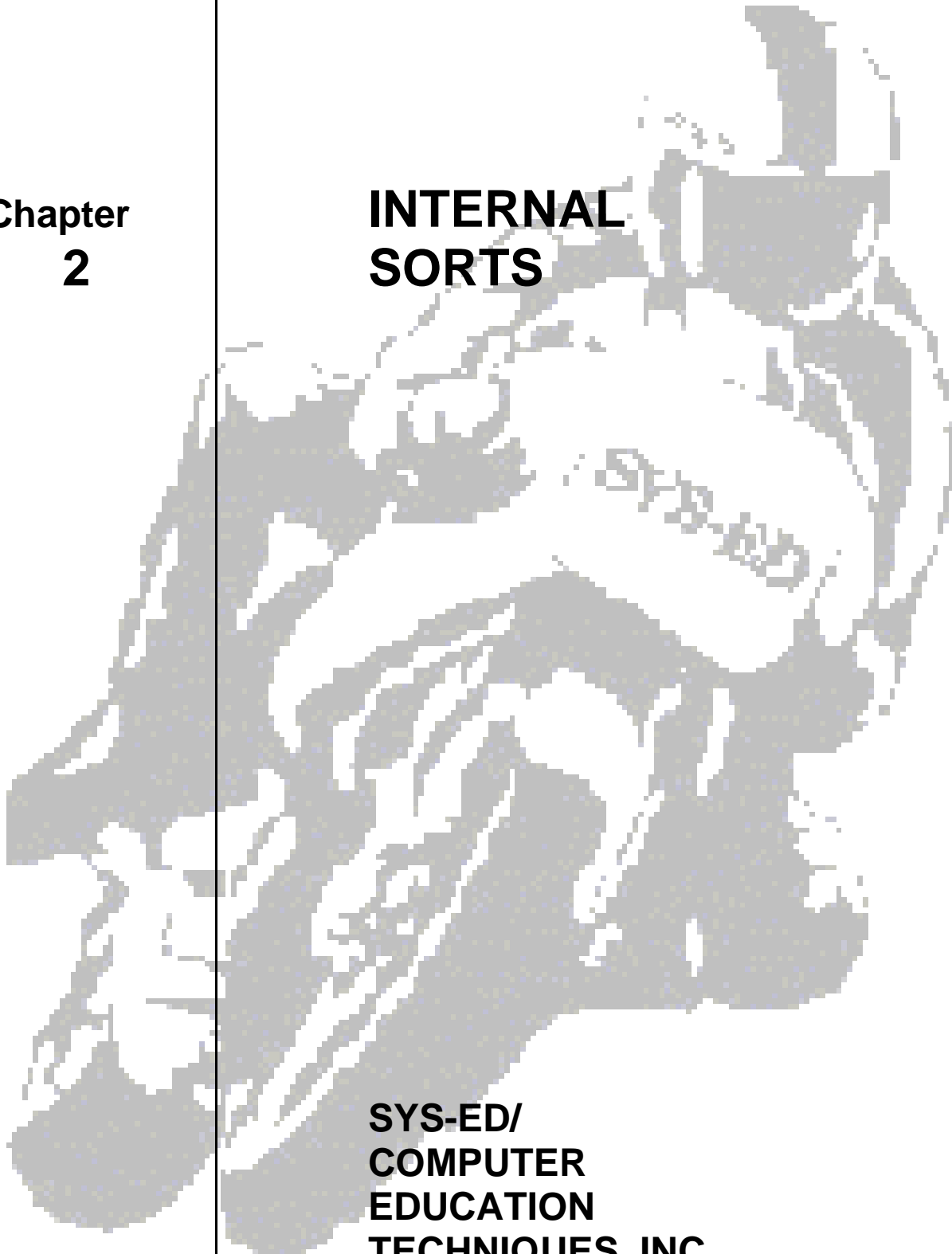


**Chapter
2**

**INTERNAL
SORTS**

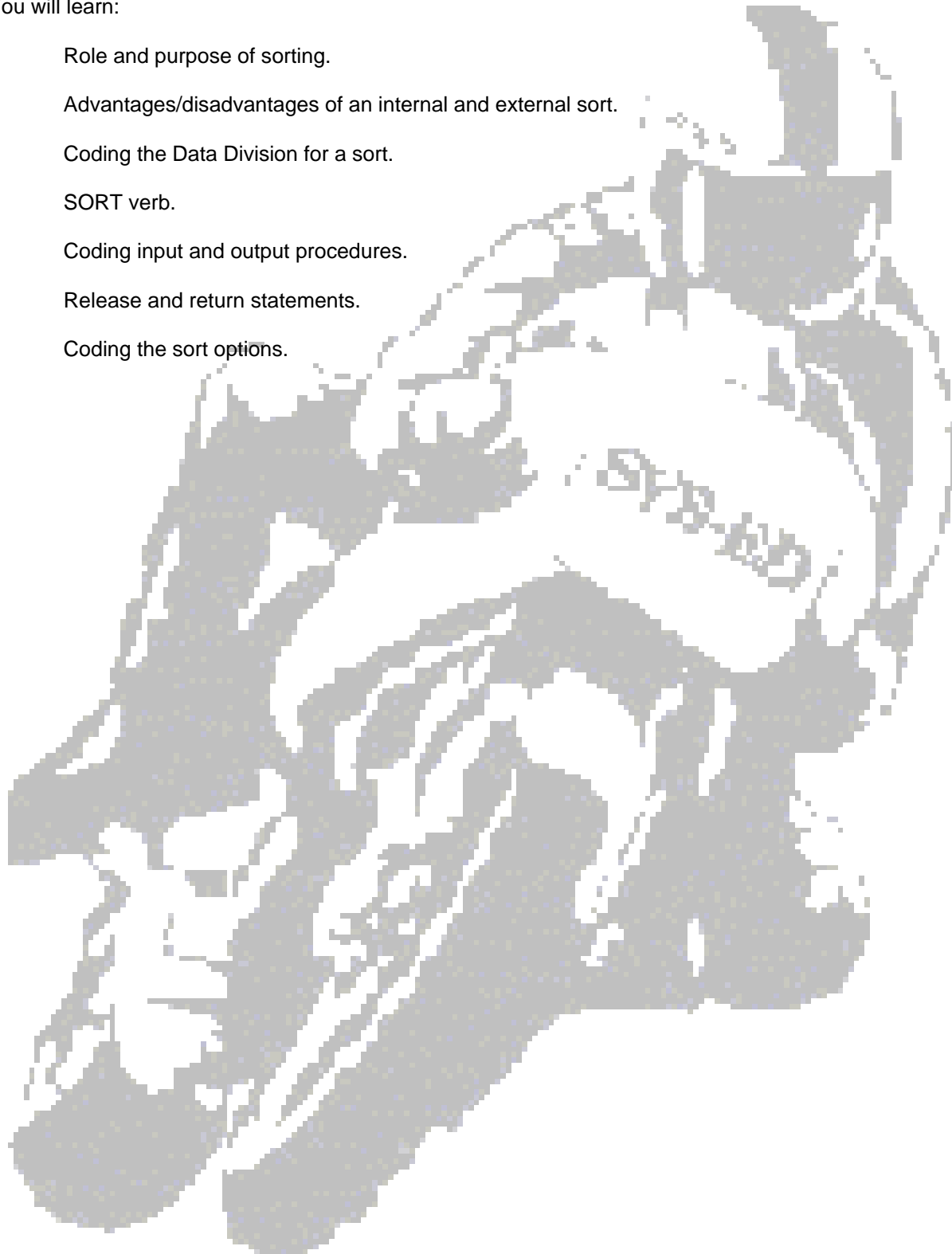


**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

You will learn:

- C Role and purpose of sorting.
- C Advantages/disadvantages of an internal and external sort.
- C Coding the Data Division for a sort.
- C SORT verb.
- C Coding input and output procedures.
- C Release and return statements.
- C Coding the sort options.



1 What is Sorting?

Sorting is a means by which data can be processed in a sequence in which it was not initially stored.

1.1 Sorting Data Files

Sorting of sequential data files is usually accomplished in one of two ways:

- C A stand-alone sort utility requires the programmer to know the parameters of the utility program.
- C The COBOL verb SORT requires the programmer to know only COBOL.

2 Environment Division Considerations

A SELECT statement must be specified for the sort file and is used to assign a name to the sort file and give a DD name to any work datasets needed.

One to eight units may be assigned for a disk sort or three to nine may be assigned for a tape sort. However, only one SELECT statement is needed.

2.1 Sort SELECT Format

```
1      SELECT sort-file-name  
2      ASSIGN TO [integer] assignment-name-1
```

3 Data Division Considerations

In order for a sort to be performed in a COBOL program, the program must include file description entries (FD's) for files that are input to or output from the sort. In addition, the sort file itself must be defined in a sort description entry (SD).

3.1 Sort File Description

1	SD	sort-file-name
2		[RECORD CONTAINS [integer-1 TO] integer-2 CHARACTERS]
3		DATA RECORD IS sort-record-name.
4	01	sort-record-name

4 Procedure Division Considerations

The Procedure Division must contain a SORT statement to describe the sort operation. Input and output sort procedure names are optional.

4.1 SORT Statement Format

```

1          | DESCENDING |
2  SORT sort-file-name ON  <      > KEY <data-name-1>
3          | ASCENDING |
4          <data-name-2> . . .
5
6          | DESCENDING |
7  ON      <      > KEY <data-name-3> . . . ] . . .
8          | ASCENDING |
9
10 | INPUT PROCEDURE IS section-name-1 [THRU section-name-2] |
11 < | USING file-name-2 |
12 | OUTPUT PROCEDURE IS section-name-3 [THRU section-name-4] |
13 < | GIVING file-name-3 |
14
15
16

```

4.2 RELEASE Statement Format

```

RELEASE sort-record-name [FROM identifier]

```

4.3 RETURN Statement Format

```

1  RETURN sort-file-name [INTO identifier]
2  AT END imperative-statement

```

5 Programming Example

5.1 Use of the COBOL SORT Verb

These examples illustrate the use of the COBOL SORT verb:

Example 1: SORT verb with the USING and GIVING options.

```
1      ID DIVISION.  
2      .  
3      ENVIRONMENT DIVISION.  
4      .  
5      INPUT-OUTPUT SECTION.  
6      FILE-CONTROL.  
7          SELECT TRANS-FILE-IN  ASSIGN TO INTRANS.  
8          SELECT TRANS-FILE-OUT  ASSIGN TO OUTTRANS.  
9          SELECT SORT-WORK       ASSIGN TO WORKFILE.  
  
10     DATA DIVISION.  
11     FILE SECTION.  
  
12     FD      TRANS-FILE-IN  
13         .  
14         .  
15     01      TRANS-IN-RECORD          PIC X(80).  
  
16     FD      TRANS-FILE-OUT  
17         .  
18         .  
19     01      TRANS-OUT-RECORD         PIC X(80).  
  
20     SD      SORT-WORK  
21     DATA RECORD IS SORT-RECORD.  
  
22     01      SORT-RECORD.  
23         05  SD-SS-NUMBER             PIC X(9).  
24         05  SD-DEPT-NO              PIC XX.  
25         05  SD-HOURS                PIC 99.  
26         05  FILLER                  PIC X(63).  
27         05  SD-TRANS-CODE           PIC XX.
```

```
28      PROCEDURE DIVISION.  
29          SORT SORT-WORK ON ASCENDING SD-DEPT-NO  
30              ON ASCENDING KEY SD-SS-NUMBER  
31              USING TRANS-FILE-IN  
32              GIVING TRANS-FILE-OUT.  
  
      STOP RUN.
```

Example 2: Sort VERB with an input and output procedure.

```
1      ID DIVISION.  
2      .  
3      ENVIRONMENT DIVISION.  
4      .  
5      INPUT-OUTPUT SECTION.  
6      FILE-CONTROL.  
7          SELECT TRANS-FILE-IN ASSIGN TO INTRANS.  
8          SELECT TRANS-FILE-OUT ASSIGN TO OUTTRANS.  
9          SELECT SORT-WORK ASSIGN TO WORKFILE.  
  
10     DATA DIVISION.  
11     FILE SECTION.  
  
12     FD      TRANS-FILE-IN  
13         .  
14         .  
15     01     TRANS-IN-RECORD.  
16         05     FILLER                PIC X(9).  
17         05     TRANS-DEPT-NO        PIC XX.  
18         05     FILLER                PIC X(69).  
  
19     FD      TRANS-FILE-OUT  
20         .  
21         .  
22     01     TRANS-OUT-RECORD          PIC X(80).  
  
23     SD      SORT-WORK  
24         DATA RECORD IS SORT-RECORD.  
  
25     01     SORT-RECORD.  
26         05     SD-SS-NUMBER          PIC X(9).  
27         05     SD-DEPT-NO            PIC XX.  
28         05     SD-HOURS              PIC 99.  
29         05     FILLER                PIC X(63).  
30         05     SD-TRANS-CODE         PIC XX.
```

```
31      WORKING-STORAGE SECTION.
32      01  CONTROL-PARAMETER.
33          05  SELECT-DEPT-NO          PIC XX.
34      01  END-FILE-SWITCH              PIC X VALUE SPACES.
35          88  END-OF-FILE              VALUE `Y'.

36      PROCEDURE DIVISION.
37          SORT SORT-WORK ON ASCENDING KEY SD-DEPT-NO
38              SD-SS-NUMBER
39              INPUT PROCEDURE 100-SELECT-RECORDS
40                  THRU 100-EXIT
41              OUTPUT PROCEDURE 500-PROCESS-SORTED-RECORDS
42                  THRU 500-EXIT.
43      STOP RUN.

44      100-SELECT-RECORDS SECTION.
45          ACCEPT CONTROL-PARAMETER.
46          IF CONTROL-PARAMETER = `NO'.
47              GO TO 100-EXIT.
48          OPEN INPUT TRANS-FILE-IN.
49          PERFORM 300-READ-TRANS-FILE THRU 300-EXIT.
50          PERFORM 200-SELECT-TRANS THRU 200-EXIT
51              UNTIL END-OF-FILE.
52          CLOSE TRANS-FILE-IN.
53          GO TO 100-EXIT.

54      200-SELECT-TRANS.
55          IF SELECT-DEPT-NO = TRANS-DEPT-NO
56              RELEASE SORT-RECORD FROM TRANS-IN-RECORD.
57          PERFORM 300-READ-TRANS-FILE.

58      200-EXIT.
59          EXIT.

60      300-READ-TRANS-FILE.
61          READ TRANS-IN-FILE
62              AT END
63              MOVE `Y' TO END-FILE-SWITCH.
```

```
64         300-EXIT.  
65             EXIT.  
  
66         100-EXIT.  
67             EXIT.  
  
68         500-PROCESS-SORTED-RECORDS SECTION.  
  
69             OPEN OUTPUT TRANS-FILE-OUT.  
70             MOVE SPACES TO END-FILE-SWITCH.  
71             PERFORM 600-RETURN-SORT-REC THRU 600-EXIT.  
72             PERFORM 700-OUTPUT-REPORT THRU 700-EXIT  
73             UNTIL END-OF-FILE.  
74             .  
75             .  
76             CLOSE TRANS-FILE-OUT.  
77             GO TO 500-EXIT.  
  
78         600-RETURN-SORT-REC.  
  
79             RETURN SORT-WORK  
80             AT END MOVE 'Y' TO END-FILE-SWITCH.  
  
81         600-EXIT.  
82             EXIT.  
  
83         700-OUTPUT-REPORT.  
84             .  
85             .  
86             PERFORM 600-RETURN-SORT-REC THRU 600-EXIT.  
  
87         700-EXIT.  
88             EXIT.  
  
89         500-EXIT.  
90             EXIT.
```

5.2 SORT with Using/Output PROC

```

ID DIVISION.
0200 PROGRAM-ID. ch21.
***** THIS PGM SORTS EMPF BEFORE PROCESSING IT *****
***** THIS PGM SORTS EMPF BEFORE PROCESSING IT *****
***** THIS PGM SORTS EMPF BEFORE PROCESSING IT *****
0300 ENVIRONMENT DIVISION.
0400 INPUT-OUTPUT SECTION.
0500 FILE-CONTROL.
0600     SELECT SORTED-EMPF ASSIGN TO SORTDD.
0600     SELECT EMPF          ASSIGN TO PAYDD.
0700
0800 DATA DIVISION.
0900 FILE SECTION.
1000 FD EMPF
1100     LABEL RECORD IS STANDARD.
1200 01 EMP-REC          PIC X(80).
1000 SD SORTED-EMPF
1100     LABEL RECORD IS STANDARD.
1200 01 SORT-REC.
1300     05 SORT-SSN      PIC X(9).
1300     05 SORT-NAME     PIC X(30).
1500     05 SORT-SKILL   PIC xxx.
1600     05 FILLER       PIC X(38).

2000 WORKING-STORAGE SECTION.
     01 WS-EOF-SWITCH   PIC X(1) VALUE 'N'.
     01 WS-PREV-SKILL   PIC xxx.

2900 PROCEDURE DIVISION.
3000 100-MAIN-MODULE.
     130-SORT.
         SORT SORTED-EMPF
             ON ASCENDING KEY XXXXXXXX
             USING XXXXXXXX
             OUTPUT PROCEDURE IS XXXXXXXXXXXXXXXX
             THRU XXXXXXXXXXXXXXXX-EXIT.
         IF XXXXXXXX NOT = ZERO
             DISPLAY ' SORT ERROR - BAD NEWS'
         END-IF
4200 STOP RUN.
*****
XXXXXXXXXXXXX SECTION.
3600
     XXXXXXXX SORTED-EMPF INTO WS-REC-IN
     AT END
3700     MOVE 'Y' TO WS-EOF-SWITCH
     END-XXXXXX

     PERFORM UNTIL WS-EOF-SWITCH = 'Y'.

         IF XXXXXXXX NOT = XXXXXXXX
             * CALCULATE TOTALS FOR EACH SKILL CODE AND PRINT IT
             END-IF
3600

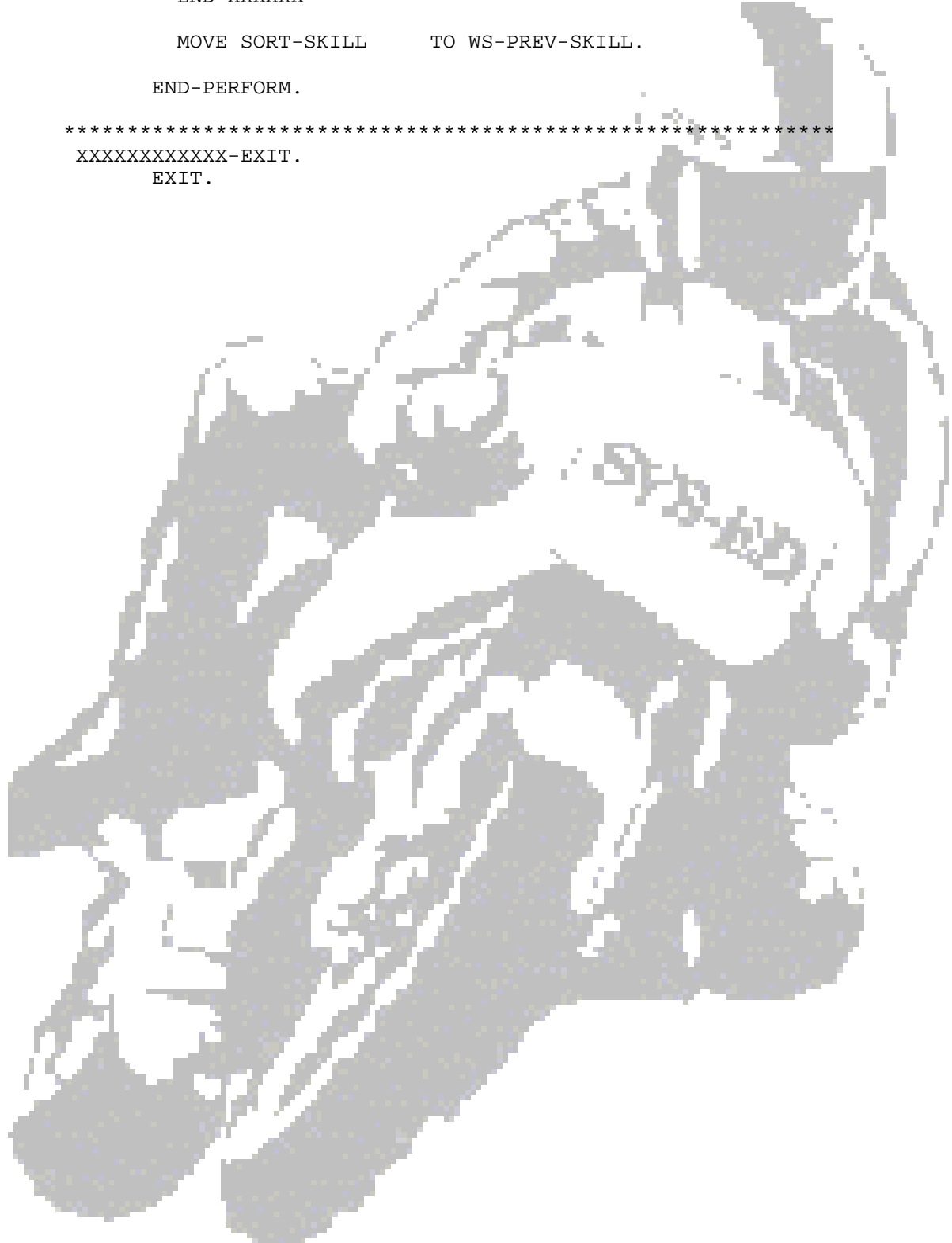
```

```
3600      XXXXXXX SORTED-EMPF INTO WS-REC-IN
          AT END
3700      MOVE 'Y' TO WS-EOF-SWITCH
          END-XXXXXX

          MOVE SORT-SKILL      TO WS-PREV-SKILL.

          END-PERFORM.

*****
XXXXXXXXXXXX-EXIT.
EXIT.
```



6 Faster Sort Facility

When using the Data Facility Sort (DFSORT) a faster sort may be achieved.

C DFSORT achieves faster sorting by doing the related input/output operation itself.

! In traditional sort packages, the standard access methods are used.

Achieving faster sort (Specifications and Limitations):

C The FASTSRT compiler option must be specified.

! There is only one input file, with no declaratives (label or exceptional error).

! For fast sorting the input, specify the USING option instead of an input procedure.

! For fast sorting the output, specify the GIVING option instead of an output procedure.

6.1 Example of Fastsort

In the most basic sort example, the relevant COBOL statements would typically be:

```

1      SELECT SORT-INPUT ASSIGN TO INFILE ....
2      SELECT SORT-OUTPUT ASSIGN TO OUTFILE ....
3
4      SORT SORT-FILE ON ASCENDING KEY sort-key
5          USING SORT-INPUT
6          GIVING SORT-OUTPUT

```

The JCL would typically be:

```

1      //jobname JOB
2      //stepname EXEC PGM=SORTPGM
3      //INFILE DD      (Input data set)
4      //OUTFILE DD     (Output data set)
5      //SYSOUT DD     SYSOUT=*
6      //SYSDBOUT DD   SYSOUT=*
7      //IGZSRTCDD DD  *      (sort control cards)
8      SMS=nnnn (SORT-MODE-SIZE)
9      OPTION      (any dfsort option)

```