From this point on, each chapter includes one practice program with a suggested solution provided to assist you in reviewing the material in the chapter. First plan and code the practice program on your own. Then check your solution against the one illustrated. A problem definition is provided that includes (1) a systems flowchart, which is an overview of the input and output, (2) record layout forms (we use different types to familiarize you with them), and (3) Printer Spacing Charts, if printed output is required. Some chapters include either an interactive program or a batch program as the practice program and some include both.

Write a batch program to write an output salary disk file from input master employee disk records. The problem definition is as follows:

**Systems Flowchart**

<table>
<thead>
<tr>
<th>IN-EMPLOYEE-FILE</th>
<th>OUT-SALARY-FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image-url" alt="Flowchart Image" /></td>
<td></td>
</tr>
</tbody>
</table>

**IN-EMPLOYEE-FILE Record Layout**

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE NAME</td>
<td>20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>SALARY</td>
<td>5</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>NO. OF DEPENDENTS</td>
<td>1</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>FICA (Soc. Sec. Tax)</td>
<td>5</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>STATE TAX</td>
<td>6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>FEDERAL TAX</td>
<td>6</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>DATE OF HIRE</td>
<td>8</td>
<td>MO (2 digits)&lt;br&gt;DA (2 digits)&lt;br&gt;YR (4 digits)</td>
</tr>
</tbody>
</table>

**OUT-SALARY-FILE Record Layout**

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPLOYEE NAME</td>
<td>20</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>SALARY</td>
<td>5</td>
<td>Alphanumeric</td>
</tr>
</tbody>
</table>

*Note: These are not COBOL field-names.*

Figure 4.4 illustrates the pseudocode for this program. Pseudocode as a planning tool will be discussed in detail in the next chapter. Look at the pseudocode planning tool first to see if you understand the logic and then try to write the program yourself. Compare your coding with the solution in Figure 4.5.

**Figure 4.4** Pseudocode for the Practice Program.

**Pseudocode**

START
Open the Files
PERFORM UNTIL no more records (ARE-THERE-MORE-RECORDS = 'NO')
Read a record
Move Input Fields to the Output Area
Write the Output Record
END-PERFORM
Close the Files
STOP
Figure 4.5  Solution to the Batch Practice Program.

IDENTIFICATION DIVISION.
PROGRAM-ID. SAMPLE.
******************************************************************************
* sample - updates a file with employee
* names and salaries *
******************************************************************************
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT IN-EMPLOYEE-FILE ASSIGN TO DATANE.
  SELECT OUT-SALARY-FILE ASSIGN TO DATANS.
******************************************************************************
DATA DIVISION.
FILE SECTION.
FD IN-EMPLOYEE-FILE.
  01 IN-EMPLOYEE-REC.
    05 IN-EMPLOYEE-NAME PIC X(20).
    05 IN-SALARY PIC X(5).
    05 IN-NO-OF-DEPENDENTS PIC X(1).
    05 IN-FICA PIC X(5).
    05 IN-STATE-TAX PIC X(6).
    05 IN-FED-TAX PIC X(6).
    05 DATE-OF-HIRE.
      10 MD PIC 9(2).
      10 DA PIC 9(2).
      10 YR PIC 9(4).
FD OUT-SALARY-FILE.
  01 OUT-SALARY-REC.
    05 OUT-EMPLOYEE-NAME PIC X(20).
    05 OUT-SALARY PIC X(5).
WORKING-STORAGE SECTION.
  01 WS-WORK-AREAS.
    05 ARE-THERE-MORE-RECORDS PIC X(3) VALUE 'YES'.
******************************************************************************
PROCEDURE DIVISION.
******************************************************************************
* 100-main-module - controls opening and closing files *
* and direction of program logic; *
* returns control to operating system *
******************************************************************************
100-MAIN-MODULE.
  OPEN INPUT IN-EMPLOYEE-FILE
  OUTPUT OUT-SALARY-FILE
  PERFORM UNTIL ARE-THERE-MORE-RECORDS = 'NO'
  READ IN-EMPLOYEE-FILE
  AT END
    MOVE 'NO' TO ARE-THERE-MORE-RECORDS
  NOT AT END
    PERFORM 200-PROCESS-RTN
  END-READ
  END-PERFORM
  CLOSE IN-EMPLOYEE-FILE
  OUT-SALARY-FILE
STOP RUN.
******************************************************************************
* 200-process-rtn - performed from 100-main-module *
* moves employee information to output *
* areas, then writes the record *
******************************************************************************
200-PROCESS-RTN.
  MOVE IN-EMPLOYEE-NAME TO OUT-EMPLOYEE-NAME
  MOVE IN-SALARY TO OUT-SALARY
  WRITE OUT-SALARY-REC.