```
1  ************************************************************;
2  *  Using Iterative DO Loops                                *;
3  ************************************************************;
4  *  Syntax                                                  *;
5  *                                                          *;
6  *    DATA output-table;                                    *;
7  *        SET input-table;                                  *;
8  *        . . .                                             *;
9  *        DO index-column = start TO stop <BY increment>;   *;
10 *           . . . repetitive code . . .                    *;
11 *           OUTPUT;                                        *;
12 *        END;                                              *;
13 *        . . .                                             *;
14 *        OUTPUT;                                           *;
15 *    RUN;                                                  *;
16 ************************************************************;
17 *  Demo                                                    *;
18 *  1) Open the PG2.SAVINGS table. Notice there are four    *;
19 *     rows representing different people. The Amount       *;
20 *     value is a monthly savings value.                    *;
21 *  2) Run the program and notice that four rows are        *;
22 *     created due to four rows being read from the input   *;
23 *     table. Also, notice how the Savings value keeps      *;
24 *     increasing for each row.                             *;
25 *  3) Fix the issue by adding an assignment statement      *;
26 *     before the DO loop to set the value of Savings to    *;
27 *     0. Run the program and notice the correct values     *;
28 *     for Savings.                                         *;
29 *  4) Add an outer DO loop to iterate through five years   *;
30 *     per each of the 12 months. Run the program and       *;
31 *     notice that you have one row per each person. Each   *;
32 *     row represents the savings after five years,         *;
33 *     assuming that savings are added each month. The      *;
34 *     value of Year is 6 and the value of Month is 13, an  *;
35 *     increment beyond each stop value.                    *;
36 *  5) Add an OUTPUT statement to the bottom of the outer   *;
37 *     DO loop. Run the program and notice that you now     *;
38 *     have 5 rows per each person (a total of 20 rows).    *;
39 *     Each row represents the savings at each of the five  *;
40 *     years.                                               *;
41 *  6) Move the OUPUT statement to the bottom of the inner  *;
42 *     DO loop. Run the program and notice that you now     *;
43 *     have 60 rows per each person (a total of 240 rows).  *;
44 *     Each row represents the savings at each year and     *;
45 *     month combination.                                   *;
46 ************************************************************;
47
48
49 data YearSavings;
50     set pg2.savings;
51     *add an assignment statement;
52
```

```
53      Savings=0;
54      do Year=1 to 5;
55          do Month=1 to 12;
56              Savings+Amount;
57              Savings+(Savings*0.02/12);
58          end;
59          output;
60      end;
61      format Savings comma12.2;
```