Summary of Lesson 6: Processing Repetitive Code

Using Iterative DO Loops

```
DATA output-table;
     . . .
     DO index-column = start TO stop <BY increment>;

          . . . repetitive code . . .

     END;
     . . .
RUN;
```

- The iterative DO loop executes statements between the DO and END statements repetitively, based on the value of an index column.

- The *index-column* parameter names a column whose value controls execution of the DO loop. This column is included in the table that is being created unless you drop it.

- The *start* value is a number or numeric expression that specifies the initial value of the index column.

- The *stop* value is a number or numeric expression that specifies the ending value that the index column must exceed to stop execution of the DO loop.

- The *increment* value specifies a positive or negative number to control the incrementing of the index column. The BY keyword and the increment are optional. If they are omitted, the index column is increased by 1.

```
DATA output-table;
     SET input-table;
     . . .
     DO index-column = start TO stop <BY increment>;

          . . . repetitive code . . .
            <OUTPUT;>

     END;
     . . .
     <OUTPUT;>
RUN;
```

- The DO loop iterates for each iteration of the DATA step.

- An OUTPUT statement between the DO and END statements outputs one row for each iteration of the DO loop.

- An OUTPUT statement after the DO loop outputs a row based on the final iteration of the DO loop. The index column will be an increment beyond the stop value.

- DO loops can be nested.

Using Conditional DO Loops

```
DATA output-table;
      SET input-table;
      . . .
      DO UNTIL | WHILE (expression);
            . . . repetitive code . . .
            <OUTPUT;>
      END;
RUN;
```

- A conditional DO loop executes based on a condition, whereas an iterative DO loop executes a set number of times.

- A DO UNTIL executes until a condition is true, and the condition is checked at the ***bottom*** of the DO loop. A DO UNTIL loop always executes at least one time.

- A DO WHILE executes while a condition is true, and the condition is checked at the ***top*** of the DO loop. A DO WHILE loop does not iterate even once if the condition is initially false.

- The expression needs to be in a set of parentheses for the DO UNTIL or DO WHILE

```
DATA output-table;
      SET input-table;
      . . .
      DO index-column = start TO stop <BY increment> UNTIL | WHILE (expression);
            . . . repetitive code . . .
      END;
      . . .
RUN;
```

- An iterative DO loop can be combined with a conditional DO loop. The index column is listed in the DO statement before the DO UNTIL or DO WHILE condition.

- For an iterative loop combined with a DO UNTIL condition, the condition is checked ***before*** the index column is incremented at the bottom of the loop.

- For an iterative loop combined with a DO WHILE condition, the condition is checked at the ***top*** of the loop and the index column is incremented at the ***bottom*** of the loop.