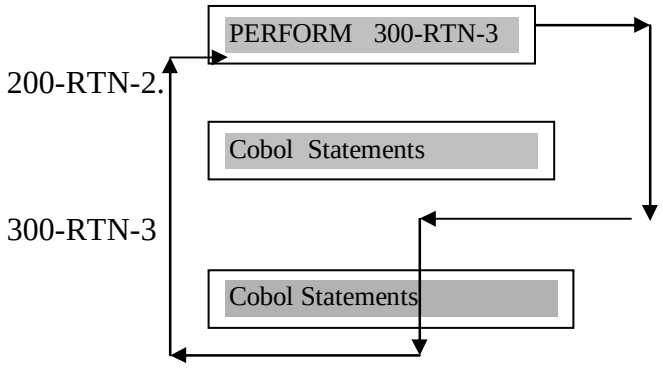
Simple Perform

The simple perform is used to for executing a specified routine from or more points in the a program.

```
PERFROM    paragraph-
```

The PERFORM statement will :

1. Execute all instructions in the named paragraph.
2. Transfer control to the next instruction in sequence, after the perform.



A simple PERFORM statement is used whenever a series of instructions in a particular paragraph is to be executed from different points in a program. An example is the printing of a heading at the top of each new page.

Example

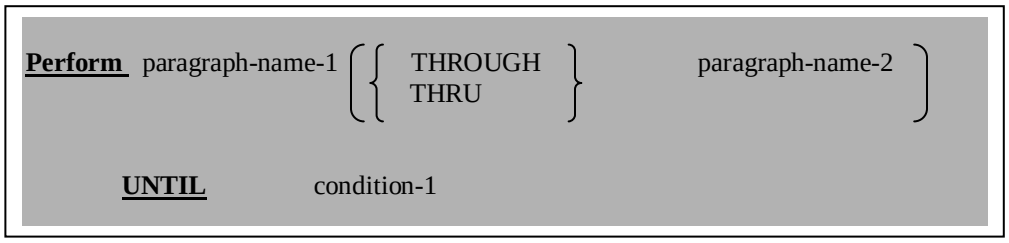
```
Procedure Division.
100-Main-Module.
        .
        .
        Perform  400-HEADING-RTN.
        .
        .
200-CALC-RTN.
        .
        .
        IF  Line-Counter = 25
                Perform  400-HEADING-RTN.
        .
        .

300-ERR-RTN.
        .
        .
        IF Line-Counter =  25
                Perform  400-HEADING-RTN.

400-HEADING-RTN.
        .
        .
```

PERFORM .. UNTIL

The format of the PERFORM UNTIL is as follows



Any simple or compound condition can be specified in a PERFORM .. UNTIL:

Example:

1. Perform 600-Error-RTN Thru 800-Error-RTN Until X = 2.

2. Perform 800-PRINT-RTN UNTIL A = B OR A = C.

3. Perform 100-Main-Module UNTIL EOF-Flag = 'YES'.

A PERFORM .. UNTIL Tests for the Condition First.

If the condition indicated in the UNTIL clause is met at the time of execution, then the named paragraph(s) will be executed 0 times.

Example:

If PERFORM 600-PROCESS-RTN UNTIL X = 3 is executed and X has 3 initially, then 600-PROCESS-RTN will not be performed at all.

Common Error to Avoid.

What is wrong with this program segment:

```
        Move 0 to Counter-A.
        PERFORM 400-Add-RTN Until Counter-A = 10.
        Write Total-Rec.
        .
        .

400-Add-RTN.
        Add Amount-In to Total.
        Read In-File At End Move 'No ' To EOF-Flag.

500-Print-RTN.
        .
        .
```

PERFORM .. TIMES

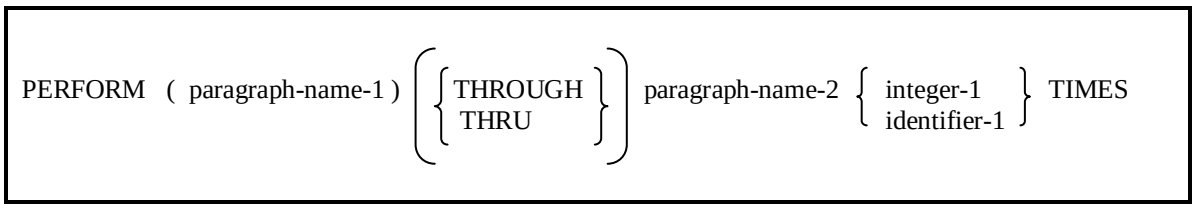
One can instruct the computer to execute a sequence of steps *a fixed number of times.*

Example 1

```
PERFORM   400-ADD-RTN   10  TIMES.
..
..

400-ADD-RTN.
        ADD  AMT1  TO  TOTALS.
        READ  IN-FILE  AT  END  MOVE  'NO ' TO  EOF-INDICATOR.
```

THE PERFORM .. TIMES FORMAT.



Example 2

ア
```
    IN-REC.
    05  NAME                    PIC   X(20).
    05  NO-OF-COPIES      PIC   9.
    .
    .
    .

PROCEDURE  DIVISION.
.
.
.
PERFORM   600-CREDIT-CARD-RTN    NO-OF-COPIES   TIMES.
.
.
```

600-CREDIT-CARD-RTN.
.
.
.

When using this format :

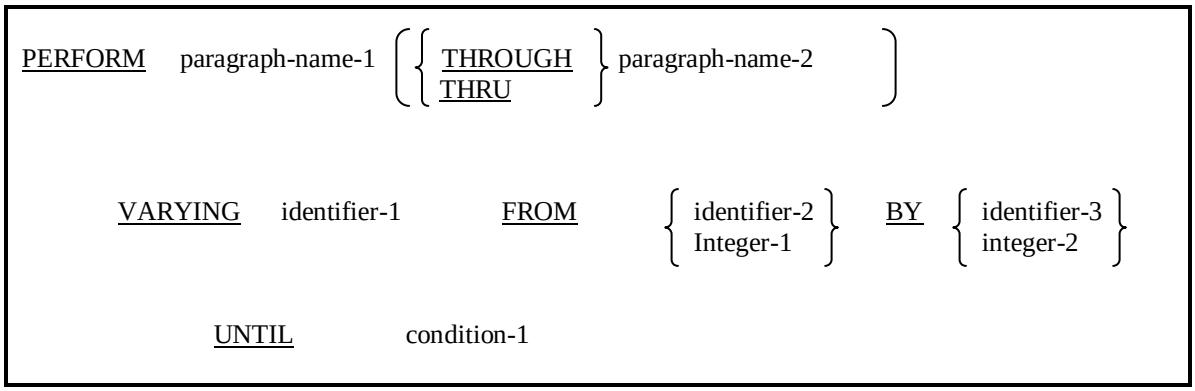
1) The identifier must be specified in the DATA DIVISION.
2) The identifier must have a *Numeric* PICTURE clause.
3) The identifier must contain ONLY numeric values or ZEROS.

In the above example, How many times does 600-CREADIT-CARD-RTN execute if the NO-OF-COPIES contain a value ZERO ???????

Practice : Find the Sum of ODD Numbers from 1 through 99 (Sum = 1 + 3 + 5 … + 99).

PERFORM ….. VARYING

The most comprehensive form of perform statement.



Example 1: Suppose One wants to sum all the ODD numbers from 1 to 1001. Using the Perform Varying, the format would be as follows :

```
Perform 300-ADD-RTN Varying Integer1 From 1 By 2 Until Integer1 > 1001.


300-ADD-RTN.
        Add Integer1 to ODD-TOTAL.
```

The Perform Varying :-

1) Initializes Integer1 to be 1.
2) Test to see if Integer1 is > 1001.
3) If Integer1 is < = 1001, Then
   i) Perform 300-ADD-RTN.
   ii) 2 is added to Integer1.
   iii) Repeat Step 2 and 3.
4) When Interg1 exceeds 1001, the process continues with the statement following the Perform statement.

Example 2 : How many times the 300-READ-RTN will be executed ????

```
Perform 300-READ-RTN Varying CTR from 1 by 1 Until CTR is Greater Than 20

Perform 300-READ-RTN Varying CTR from 0 by 1 Until CTR is = 20
```

Example 3 :

```
Perform 100-RTN1 Varying DateX From 1900 BY 10 Until DateX > 1990.

Perform 100-RTN1 Varying Counter From 10 BY -1 Until Counter = 0.
```