COBOL - LOOP STATEMENTS

There are some tasks that need to be done over and over again like reading each record of a file till its end. The loop statements used in COBOL are:

- Perform Thru
- Perform Until
- Perform Times
- Perform Varying

Perform Thru

Perform Thru is used to execute a series of paragraph by giving the first and last paragraph names in the sequence. After executing the last paragraph control is returned back.

In-line Perform

Statements inside the PERFORM will be executed till END-PERFORM is reached.

Syntax

Following is the syntax of In-line perform:

```
PERFORM
   DISPLAY 'HELLO WORLD'
END-PERFORM.
```

Out-of-line Perform

Here, a statement is executed in one paragraph and then the control is transferred to other paragraph or section.

Syntax

Following is the syntax of Out-of-line perform:

```
PERFORM PARAGRAPH1 THRU PARAGRAPH2
```

Example

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.

PROCEDURE DIVISION.
   A-PARA.
   PERFORM DISPLAY 'IN A-PARA'
   END-PERFORM.
   PERFORM C-PARA THRU E-PARA.

   B-PARA.
   DISPLAY 'IN B-PARA'.
   STOP RUN.

   C-PARA.
   DISPLAY 'IN C-PARA'.

   D-PARA.
   DISPLAY 'IN D-PARA'.

   E-PARA.
```

```
   DISPLAY 'IN E-PARA'.
```

JCL to execute the above COBOL program.

```
//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEP1 EXEC PGM=HELLO
```

When you compile and execute the above program, it produces the following result:

```
IN A-PARA
IN C-PARA
IN D-PARA
IN E-PARA
IN B-PARA
```

Perform Until

In perform until', a paragraph is executed until the given condition becomes true. With test before is the default condition and it indicates that the condition is checked before the execution of statements in a paragraph.

Syntax

Following is the syntax of perform until:

```
PERFORM A-PARA UNTIL COUNT=5

PERFORM A-PARA WITH TEST BEFORE UNTIL COUNT=5

PERFORM A-PARA WITH TEST AFTER UNTIL COUNT=5
```

Example

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.

DATA DIVISION.
   WORKING-STORAGE SECTION.
   01 WS-CNT PIC 9(1) VALUE 0.

PROCEDURE DIVISION.
   A-PARA.
   PERFORM B-PARA WITH TEST AFTER UNTIL WS-CNT>3.
   STOP RUN.

   B-PARA.
   DISPLAY 'WS-CNT : 'WS-CNT.
   ADD 1 TO WS-CNT.
```

JCL to execute the above COBOL program.

```
//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEP1 EXEC PGM=HELLO
```

When you compile and execute the above program, it produces the following result:

```
WS-CNT : 0
WS-CNT : 1
WS-CNT : 2
WS-CNT : 3
```

Perform Times

In 'perform times', a paragraph will be executed the number of times specified.

Syntax

Following is the syntax of perform times:

```
PERFORM A-PARA 5 TIMES.
```

Example

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.

PROCEDURE DIVISION.
   A-PARA.
   PERFORM B-PARA 3 TIMES.
   STOP RUN.

   B-PARA.
   DISPLAY 'IN B-PARA'.
```

JCL to execute the above COBOL program:

```
//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEP1 EXEC PGM=HELLO
```

When you compile and execute the above program, it produces the following result:

```
IN B-PARA
IN B-PARA
IN B-PARA
```

Perform Varying

In perform varying, a paragraph will be executed till the condition in Until phrase becomes true.

Syntax

Following is the syntax of perform varying:

```
PERFORM A-PARA VARYING A FROM 1 BY 1 UNTIL A=5.
```

Example

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.

DATA DIVISION.
   WORKING-STORAGE SECTION.
   01 WS-A PIC 9 VALUE 0.

PROCEDURE DIVISION.
   A-PARA.
   PERFORM B-PARA VARYING WS-A FROM 1 BY 1 UNTIL WS-A=5
   STOP RUN.

   B-PARA.
   DISPLAY 'IN B-PARA ' WS-A.
```

JCL to execute the above COBOL program:

```
//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEP1 EXEC PGM=HELLO
```

When you compile and execute the above program, it produces the following result:

```
IN B-PARA 1
IN B-PARA 2
IN B-PARA 3
IN B-PARA 4
```

GO TO Statement

GO TO statement is used to change the flow of execution in a program. In GO TO statements transfer goes only in the forward direction. It is used to exit a paragraph. The different types of GO TO statements used are as follows:

Unconditional GO TO

```
GO TO para-name.
```

Conditional GO TO

```
GO TO para-1 para-2 para-3 DEPENDING ON x.
```

If 'x' is equal to 1, then the control will be transferred to first paragraph and if 'x' is equal to 2, then the control will be transferred to the second paragraph, and so on.

Example

```
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.

DATA DIVISION.
   WORKING-STORAGE SECTION.
   01 WS-A PIC 9 VALUE 2.

PROCEDURE DIVISION.
   A-PARA.
   DISPLAY 'IN A-PARA'
   GO TO B-PARA.

   B-PARA.
   DISPLAY 'IN B-PARA '.
   GO TO C-PARA D-PARA DEPENDING ON WS-A.

   C-PARA.
   DISPLAY 'IN C-PARA '.

   D-PARA.
   DISPLAY 'IN D-PARA '.
   STOP RUN.
```

JCL to execute the above COBOL program:

```
//SAMPLE JOB(TESTJCL,XXXXXX),CLASS=A,MSGCLASS=C
//STEP1 EXEC PGM=HELLO
```

When you compile and execute the above program, it produces the following result:

```
IN A-PARA
IN B-PARA
IN D-PARA
```