

Chapter
1

INTRODUCTION

*Get on the
Fast Track!*



TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

You will learn:

- C Features of DYL-280.
- C Differences between procedural and non-procedural languages.
- C Categories and summary of DYL-280 statements.
- C Syntax rules.

1 What are 4 GLs

A fourth generation language will serve to:

- C Speed up application building.
- C Reduce maintenance cost.
- C Minimize debugging.
- C Generate bug free code from high level expressions.
- C User friendly - Minimize learning curve.
- C Non Procedural Language.

2 Functions of DYL-280

DYL-280 can perform the following functions:

- C General purpose report generation.
- C Computations.
- C Selection criteria.
- C Sort files.
- C Merge files.
- C Edit files.
- C Built in utilities.
- C Label generation.

3 DYL-280 Advantages

The advantages in using DYL-280 include:

- C ease of use.
- C reduction of application backlog.
- C user programmability.
- C simple-to-use, English like commands.
- C flexible.
- C works with QSAM and VSAM.

4 Types of Commands

The DYL-280 commands fall into the following categories:

- C computational.
- C writing detail and summary information.
- C setting up title/footering.
- C data set and descriptions.
- C conditional expressions.
- C sorting the file.
- C flow of control.

5 Data Types

DYL-280 uses the following data types:

| | |
|----|----------------|
| CH | Character |
| NU | Zoned Numeric |
| PD | Packed Decimal |
| BI | Binary |

6 Program Layout

The program layout in DYL-280 is as follows:

- C OPTION statement.
- C REPORT statement.
- C FILE command.
- C WORKAREA CONTROL.
- C SORT.
- C Detail time processing.
- C Subroutines.
- C ON END.
- C ON CHANGE.
- C ON FINAL.
- C TITLE and title modifications.
- C FIN.

7 FILE Organization

DYL 280 file organization is as follows:

| QSAM | Sequential Access |
|-------------|--------------------------|
| C | Fixed |
| C | Variable |
| C | Spanned |
| C | Undefined |

| BDAM | Direct |
|-------------|------------------------------|
| C | Sequential and Direct Access |

| VSAM | KSDS, ESDS, RRDS |
|-------------|--------------------------|
| C | Fixed |
| C | Variable (KSDS and ESDS) |
| C | Sequential and Direct |

| Library | PDS or BPAM |
|----------------|--------------------------|
| C | Fixed or Variable |
| C | Full Directory Retrieval |

8 General Rules

Free Form

Using the free form option, the statements can be coded in columns 1-72. Some older programs use the fixed form option.

Comment

Coding an asterisk(*) in column 1 indicates that the entire line is a comment.

Anything after a semicolon(;) is also considered a comment unless it is in a literal.

Literals

There are three types of literals:

| | |
|--------------|--|
| Alphanumeric | Maximum size is 255 characters. |
| Numeric | Maximum size is 20 characters. Numeric constants can include a minus symbol in front of the number and can have an embedded decimal point. |
| Hexadecimal | The value is in the form X'nn' |

Dataname

Data names for variables can be 2-50 characters long for fields in the record or workarea. Data names for files, tables, and parms can be 2-10 characters.

Filename

Maximum size is 8 characters.

Tag or Label

Maximum size is 10 characters.

9 Automatic Cycle

```
OPEN FILES
DO UNTIL END-OF-FILE
    READ INPUT FILE
    IF END-OF-FILE
        FINISH REPORT
        ON CHANGE PROCESSING
        ON FINAL PROCESSING
        PRINT RUN STATISTICS
        CLOSE FILES
        TERMINATE
    ELSE
        PROCESS STATEMENTS
        PRINT
        WRITE
    ENDIF
ENDUNTIL
```

10 FILE Command

```
FILE filename recfm size blocksize
      {INPUT|OUTPUT}
      SEQUENTIAL
      [FROM filename]
      COUNT Cntr
      LENGTH Len
```

| | |
|---------------|--|
| Filename | DDNAME of the file. |
| Recfm | Record Format F Fixed FB Fixed Block V Variable VB Variable Block U Undefined |
| Size | Logical record size. |
| Blocksize | Physical record size. |
| Len | Length of current record. Mainly used for variable length records. |
| Cntr | Current record count. |
| FROM filename | Where record came from. |