

**Chapter
2**

**DECLARING
DATA ITEMS AND
REPRESENTATION**

*Get on the
Fast Track!*



TM

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

Objectives

You will learn:

- C How to declare PL/I variables with scale, base and mode.
- C Internal representation of numeric, character and varying character strings.
- C Differences between the number types and which types to use for a particular circumstance.
- C Declaring and using bit strings.
- C How to initialize a variable in the DECLARE statement.
- C How to format data items for printing or for writing to a dataset.
- C How to code the PICTURE specification in a DECLARE statement.

1.1 Fixed Decimal Declare

DCL DOLLARS_AND_CENTS	FIXED DECIMAL (15,2);
DCL PERCENT	FIXED DECIMAL (7,4);
DCL NET_PROFIT	FIXED DEC (13,0);

1.2 Fixed Binary Declare

DCL TABLE_INDEX	FIXED BINARY (15,0);
DCL VERY_HIGH_COUNTER	FIXED BIN (31,0);

2 Floating Point Declaration

```
DCL identifier FLOAT DECIMAL(p);
```

↑ Precision

Example:

```
DCL SPEED FLOAT DECIMAL(6);
```

This declare is used to reserve a floating point number with six significant digits being maintained. Since precision is less than seven digits, short form floating point is used (4 bytes).

Example:

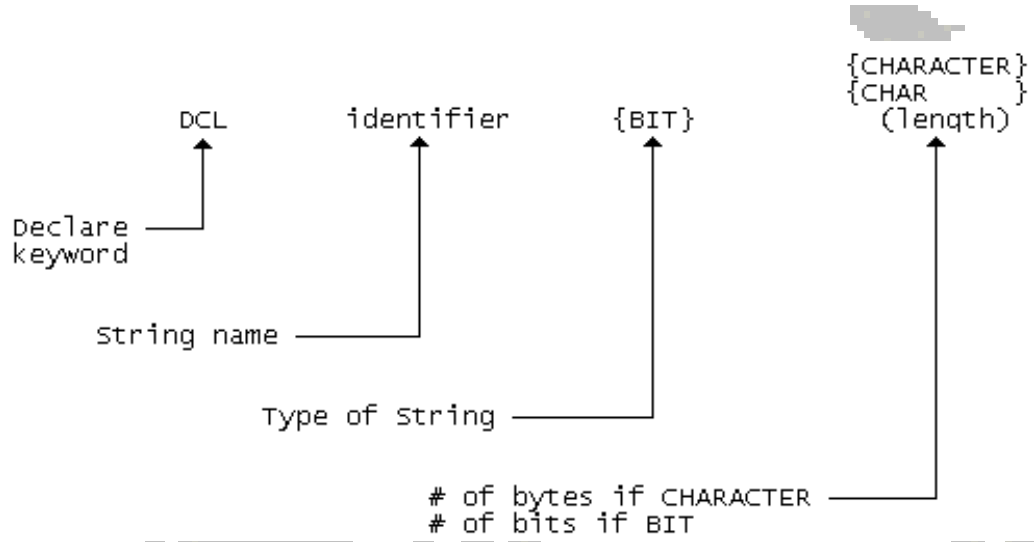
```
DCL VERY_PRECISE_NUMBER FLOAT DEC(16);
```

This declare is used to reserve a floating point number that requires more than 6 significant digits to be maintained. Sixteen digits of precision are reserved for this variable. Long form floating point is used (8 bytes).

3 Arithmetic Data Types

Data Type	Use Precision	Maximum Value	Used for
FIXED BINARY	(15,0) (31,0)	32,767 2,147,483,647	counting. array indexing. representing integers.
FIXED DECIMAL	(t,f) where: t=total digits f=fractional digits and t should be an odd	Up to 15 digits	numbers with fractional parts. data where total accuracy is required (e.g. Payroll). larger numbers FIXED BINARY cannot be accommodated.
FLOAT DECIMAL	3 sizes: (6) Short (16) Long (33) Extended	10^{75}	extremely large or small numbers. most modeling and scientific programming. convenience (PL/I will maintain the significant digits).

4 Declaration of Strings



Examples:

```
DCL NAME CHAR(20);
DCL TRUE BIT (1);
DCL CITY CHARACTER(16);
```

5 Varying Length Strings

Format:

```
DCL NAME CHAR(30) {VARYING}
                        {VAR}
```

- C Strings may have the VARYING attribute.
- C Allows the string to take on different lengths.
- C Length can vary between 0 and maximum specified in DECLARE.
- C PL/I internally tracks the string length automatically. It uses 2 extra bytes (halfword binary) added to the front of the string's storage to store the length.

6 Initial Attribute

The initial attribute is used to give starting values to variables when storage is assigned.

Format:

```
DCL identifier attribute(s)      {INITIAL}      (value or expt.);
                                {INIT}
```

Examples:

DCL NAME CHAR(10)	INIT('ALBERT'); [padded with 4 blanks]
DCL TOWN CHAR(20)	VAR INIT ('AFTON'); [stores length]
DCL FALSE BIT(1)	INIT ('O'B);
DCL ATTR BIT(4)	INIT ('101101'B); [truncates rightmost 2 bits]
DCL DOLLAR_AND_CENTS	FIXED DEC(7,2) INIT (4999.99);
DCL LARGE_NUMBER	FLOAT DEC(16) INIT (1234560000000789);
DCL RCD_COUNT	FIXED BIN(15,0) INIT (9 or FIXED BIN(15,0) INIT (1001B);

7 String Data

DATA TYPE	STORED	LENGTH
CHARACTER_STRING	1 CHARACTER PER BYTE	0 TO 32767 CHAR's
BIT_STRING	8 BITS PER BYTE	0 TO 32767 BITS

DCL NAME CHAR(20), MASK BIT(17)

String data may have the varying attribute in which case the actual length at any particular time is the length of the last data item assigned to it.

```
DCL NAME CHAR(20) VARYING;
NAME = 'JON';           /*LENGTH IS 3 BYTES */
NAME = 'JONATHON';     /*LENGTH IS 8 BYTES */
```

8 Exercise

AVERAGE:

```

PROCEDURE OPTIONS(MAIN);

    DECLARE (A,B,C,D,E) FIXED(4,1);

    GET LIST (A,B,C,D,E);

    MEAN = (A+B+C+D+E) / 5;

    PUT LIST('AVERAGE IS' , MEAN);

END AVERAGE;
    
```

Explicitly Declared Identifiers	Implicitly Declared Identifiers

9 Declaring Identifiers with the Bit Attribute

```
DECLARE
    YES          BIT(1),
    NO           BIT(1),
    MORE_TRANSACTIONS BIT(1),
    SKILLS       BIT(16);
YES = '1'B;
NO = '0'B;
MORE_TRANSACTIONS = YES;
SKILLS = '1100111000111100'B;
SKILLS = (8)'0B' ** (8)'1'B;
```

- C Pad on right with zeros if bit-string is less than the declared length.
- C Truncate on right if bit-string is greater than the declared length.

10 Initial Attribute

The initial attribute causes an identifier to be set (initialized) to a specified value before execution of other PL/I statements in a program.

```
DCL TITLE          CHAR(10) INITIAL('SIT GEORGE');
DCL DELETE         CHAR CHAR(1) INIT(' ');
DCL SUM           FIXED(5) INIT(0);
DCL PI            FIXED(6,5) INIT(3.14159);
/* NOTE ABBREVIATION */
DCL (A,B,C) FIXED(3) INIT(0);
DCL ON            BIT(1) INIT('1'B);
DCL OFF           BIT(1) INIT('0'B);
```

11 Picture Attribute

FOR DEFINING
AN ARITHMETIC
FIELD

```
DCL AMOUNT PICTURE'999V99';  
GET LIST(AMT);
```

Source Data

12345

Internal Result

123.45

FOR EDITING
DATA

```
DCL PAY FIXED(5,2);  
DCL EDIT_PAY PICTURE'$ZZZZV.99';  
EDIT_PAY = PAY;  
/* EDITING TAKES PLACE */
```

12 Picture Characters

9	ANY DECIMAL DIGIT
V	VIRTUAL DECIMAL POINT (CAUSES DECIMAL POINT ALIGNMENT)
.	DECIMAL POINT INSERT (NOT NECESSARILY RELATED TO DECIMAL POINT ALIGNMENT)
-	NEGATIVE (USUALLY USED ON OUTPUT PICTURES)
9	POSITIVE (USUALLY USED ON OUTPUT PICTURES)
s	SIGN--POSITIVE OR NEGATIVE (USED FOR INPUT AND OUTPUT)
X	ANY ALPHANUMERIC CHARACTER
A	ANY ALPHABETIC CHARACTER
\$	INSERT DOLAR SIGN
0	INSERT ASTERISK
,	INSERT COMMA
B	INSERT BLANK CHARACTER (USUALLY USED ON OUTPUT PICTURES)
CR	CREDIT SYMBOL (CR FOR NEGATIVE FIELDS, bb FOR POSITIVE FIELDS)
DB	DEBIT SYMBOL (DB FOR NEGATIVE FIELDS, bb FOR POSITIVE FIELDS)
Z	ZERO SUPPRESS (INPUT: BLANKS, INTERPRETED AS ZEROS, OUTPUT: LEADING ZEROS REPLACED BY BLANKS.)
/	INSERT SLASH

SOURCE DATA	PICTURE SPECIFICATION	RESULTING DATA
00000	'99999' 'ZZZZ9'	00000 bbbb0
00.00	'99V.99' 'ZZV.ZZ' 'ZZV.99' 'ZZ.V99'	00.00 bbbbbb bb.00 bbb00
0123.45	'9999V.99' 'ZZZZV.99' 'Z,ZZZV.99BBB'	0123.45 b123.45 bb123.45bbb
070784	'99/99/99' 'Z9/99/99' 'ZZ/ZZ/ZZ(5)B'	07/07/84 b7/07/84 b7/b7/84bbbb
001200.75	'\$ZZZ,ZZZV.99' '\$\$\$\$,\$\$\$V.99cr'	\$bb1,200.75 bb\$1,200.75bb
-123.45	'999V.99' 's999V.99' '999V.99BDB' '999V.99CR' '999V.99S' '(6)A'	123.45 -123.45 123.45bDB 123.45CR 123.45- ABCbbb

13 Declare Statement Formats

FORMAT 1:

```
DCL      AMT_OF_PURCHASE  FIXED DEC(7,2)   INIT(0);
DCL      AMT_OF_COST      FIXED DEC(7,2)   INIT(0);
DCL      AMT_RECEIVED     FIXED DEC(7,2)   INIT(0);
```

FORMAT 2:

```
DCL      AMT_OF_PURCHASE  FIXED DEC(7,2)   INIT(0),
          AMT_OF_COST      FIXED DEC(7,2)   INIT(0),
          AMT_RECEIVED     FIXED DEC(7,2)   INIT(0);
```

FORMAT 3 (FACTORING):

```
DCL      (AMT_OF_PURCHASE,
          AMT_OF_COST,
          AMT_RECEIVED)   FIXED DEC(7,2)   INIT(0);
```

FORMAT 4 (FACTORING):

```
DCL      (AMT_OF_PURCHASE  INIT(300.00),
          AMT_OF_COST      INIT(250.00),
          AMT_RECEIVED     INIT(0)        FIXED DEC(7,2);
```