บทที่ 3
ข้อมูล
(Data items)

ข้อมูล ในโปรแกรม PL/I-80 อาจจะเป็นค่าคงที่ (constants) หรือ ตัวแปร (variable), ค่าคงที่ เป็น data item ซึ่งมูลค่าของมันจะไม่มีการเปลี่ยนแปลง ระหว่างการ execute โปรแกรม ในขณะที่ ค่าของตัวแปรอาจมีการเปลี่ยนแปลงระหว่าง การ execute ข้อมูลต่าง ๆ ในโปรแกรมจะมีคุณสมบัติ (properties) เพิ่มเติม เฉพาะ เช่น ช่วงจำกัดของค่า subscript (range of subscript values), การปฏิบัติการ (operation) ซึ่งอาจจะต้องมี, หรือจำนวนเนื้อที่หน่วยความจำที่ต้องใช้, คุณสมบัติเหล่านี้ เรียก attributes ซึ่งจะถูกกำหนดให้กับตัวแปรในคำสั่ง DECLARE หรือในบางกรณี เช่น ค่าคงที่จะถูกกำหนดโดยระบบคอมพิวเตอร์ (by system default)

ตัวแปรข้อมูล (data variable) อาจจะหมายถึง ข้อมูลตัวเดียว (single data item), หรือ โครงสร้าง (structures) หรือ อะเรย์ (arrays) ซึ่งหมายถึง ข้อมูลกลุ่ม, ข้อมูลตัวเดียว อาจจะเป็นตัวแปร หรือ ค่าคงที่ เรียกว่า scalar ใน PL/I-80 รูปแบบของข้อมูลแบ่งออกเป็น 6 ชนิดคือ

arithmetic

string

pointer

label

entry

file data

ในตัวข้อต่อไปนี้ จะได้กล่าว ในรายละเอียดของข้อมูลแต่ละชนิด

3.1 ข้อมูลเลขคณิต (Arithmetic data)

PL/I-80 แบ่งข้อมูลตัวเลข ออกเป็น 3 ชนิดคือ FIXED BINARY, FLOAT BINARY, และ FIXED DECIMAL ข้อมูลตัวเลขแต่ละตัว จะมีความถูกต้อง (precision) และค่าของ scale โดยการให้ค่าคงที่จำนวนเต็ม p และ q อยู่ในวงเล็บ ความถูกต้อง

p หมายถึง จำนวนเลขฐานสิบ หรือ จำนวนเลขฐานสอง ซึ่งประกอบเป็นข้อมูลตัวนั้น และ scale q หมายถึง จำนวนตัวเลข ทางขวามือของจุดทศนิยม หรือจุดทวินิยม (binary point) ความถูกต้องและ scale ของตัวแปรแต่ละตัว สามารถกำหนดภายนอก (explicitly) เมื่อตัวแปรตัวนั้นถูก declared หรือกำหนดภายใน (implicitly) โดยกฎการ default

3.1.1 Fixed binary ตัวแปรที่ declare เป็น FIXED **BINARY**[(p)] หมายถึง เลขจำนวนเต็ม ซึ่งมี เลขฐานสอง p ตัว, ช่วงของ p คือ

1 <= p <= 15

เนื่องจากข้อมูลชนิดนี้ ถูกเก็บภายในหน่วยความจำด้วยรูปแบบ two's complement ช่วงของเลข FIXED BINARY แต่ละตัวอยู่ระหว่าง -32768 ถึง 32767 การจัดสรรเนื้อที่หน่วยความจำให้กับเลข FIXED BINARY ขึ้นอยู่กับตัว p, ถ้า p<=7 จะมีเพียง 1 byte เท่านั้นที่จัดเนื้อที่ ให้กับข้อมูลตัวนั้น ถ้าเป็นกรณีอื่น ๆ เครื่องจะจัดเนื้อที่ให้ 2 bytes, default precision คือ (15), การกำหนดค่าให้กับตัวแปรชนิด FIXED BINARY ออกนอกช่วงที่ถูกต้อง จะทำให้ได้ผลลัพธ์ที่ไม่ทราบค่า (undefined)

ค่าคงที่ชนิด FIXED BINARY เขียนเป็นเลขจำนวนเต็มฐานสิบ ค่าคงที่ซึ่งถือว่าเป็น ข้อมูลชนิด FIXED BINARY ก็ต่อเมื่อ มันปรากฏในตัวโปรแกรม ซึ่งต้องการค่า fixed binary เช่น subscripts หรือ arithmetic operation ที่เกี่ยวกับข้อมูลชนิด FIXED BINARY อื่น ๆ นอกเหนือจากนี้ เครื่องจะ default ให้เป็น FIXED DECIMAL, การเปลี่ยนรูป (conversion) จากข้อมูลชนิดอื่นเกิดขึ้นพร้อมกับการตัด (trncation) (ดู บทที่ 7 เรื่อง กฎการเปลี่ยนรูปข้อมูล)

ตัวอย่าง

```
DECLARE  I  FIXED  BINARY;
     I = 1.99;
```

หมายถึง ตัวแปร I จะมีค่าเป็น 1

การ declare ตัวแปรด้วย FIXED, BINARY หรือ FIXED BINARY มีความหมายอย่างเดียวกับการ declare ด้วย FIXED BINARY(15)

3.1.2 Fixed decimal

นอกจากตัวเลขต่าง ๆ ที่ใช้ในรูปของ FIXED BINARY, ค่าคงที่จำนวนเต็มฐานสิบทั้งหมด ซึ่งจะมีจุดทศนิยม หรือไม่มีก็ตาม เครื่องจะ default ให้เป็น FIXED DECIMAL ตัวแปรที่ declare เป็น FIXED DECIMAL[(p [,q])] เป็นเลขฐานสิบ มีเครื่องหมาย 1 ตัว, เลขฐานสิบทั้งหมด p ตัว และมีเลข q ตัวอยู่ทางขวามือของจุดทศนิยม ช่วง r ของเลข FIXED DECIMAL คือ

$$-10^{**}(p-q) < r < 10^{**}(p-q)$$

เมื่อ $1 <= p <= 15$ และ

$0 <= q <= p$

เครื่องจะ default precision และ scale เป็น (7,0), default precision และ scale ของค่าคงที่ฐานสิบ ถูกกำหนดโดยรูปแบบของค่าคงที่ โดยตัวมันเอง

ตัวอย่าง

3.25 default เป็น (3,2)

302 default เป็น (3,0)

เลขฐานสิบแต่ละตัวเก็บภายในหน่วยความจำ ในรูปแบบของ packed BCD, ค่าซึ่งมี scale มากกว่า ตัวแปรชนิด FIXED DECIMAL จะถูกตัด (truncated) ถ้าถูกกำหนดให้กับตัวแปรนั้น และถ้าค่าซึ่งมีเลขนัยสำคัญ (significant digit) มากกว่า จำนวนเลขซ้ายมือของจุดทศนิยมที่เรากำหนดให้กับตัวแปร จะเกิดความผิดพลาด (error) ชนิด FIXED OVERFLOW

3.1.3 Float binary

ตัวเลข FLOAT BINARY แต่ละตัวแบ่งออกเป็น 2 ส่วน คือ fractional part (GO mantissa) หมายถึง เลขนัยสำคัญ ของตัวเลขนั้น และ exponential part ซึ่งหมายถึง scale factor, ตัวแปรชนิด declare เป็น FLOAT BINRY(p) จะมีเครื่องหมาย s, integer exponent e และเลขฐานสอง p ตัวซึ่งหมายถึง fractional part ของตัวเลขนั้น, ช่วงของขนาด (magnitude) ของเลข FLOAT BINARY r โดยประมาณคือ

5.88*10**-39 <= r <= 3.40*10**3

ช่วงของ p คือ

1 <= p <= 24

default precision ของ p คือ 24

ค่าคงที่ชนิด FLOAT BINARY เขียนในรูปสัญลักษณ์ทางวิทยาศาสตร์ (scientific notation) ประกอบด้วย เลขฐานสิบ อาจจะมีจุดทศนิยม หรือไม่มีก็ได้ แล้วตามด้วยตัวอักษร E (ตัวใหญ่ หรือ ตัวเล็ก ก็ได้) แล้วตามด้วย decimal integer exponent ซึ่งจะมีเครื่องหมาย (sign) หรือไม่มีก็ได้เช่นกัน

ตัวอย่าง

A = 2.332;

หมายถึง A มีค่าเท่ากับ 230, การ declare ตัวแปรให้เป็น FLOAT จะมีความหมายอย่างเดียวกับ การ declare ให้เป็น FLOAT BINARY

2.1.4 Arithmetic built-in functions

เพื่อเป็นการเพิ่ม ตัวปฏิบัติการเลขคณิต (arithmetic operators) PL/I-80 ได้กำหนด arithmetic บิลท์-อิน ฟังก์ชันต่อไปนี้ ให้เป็นส่วนหนึ่งในภาษานี้

ARS	DECIMAL	MAX	TAN
ACOS	DIVIDE	MIN	TAND
ASIN	BXP	MOD	TANH
ATAN	FIXED	ROUND	TRIJNC
BINARY	FLOAT	SIGN	
CEIL	FLOOR	SIN	
COS	LOG	SIND	
COSD	LOG10	SINK	
COSH	LOG2	SQRT	

สำหรับรายละเอียดเกี่ยวกับ บิลท์-อิน ฟังก์ชัน ข้างต้นนี้ ให้ดูในบทที่ 12

3.2 String data

ข้อมูลชนิด string ใน PL/I-80 แบ่งออกเป็น 2 ชนิดคือ ข้อมูลชนิด character string และ ข้อมูลชนิด bit string, ค่าของ character string เป็น sequence of ASCII characters, รวมทั้ง empty หรือ null sequence ส่วนค่าของ bit string เป็น non-empty sequence of bits ความยาวของ string คือจำนวนตัวอักขระ หรือจำนวนบิท (bit) ใน string นั้น รายละเอียดของ กฎที่เกี่ยวกับการใช้ข้อมูลชนิด string มีดังนี้

3.2.1 Character string data

ตัวแปรซึ่ง declare เป็น CHARACTER(n) หมายถึง character string มีความยาวเท่ากับ n เมื่อ n เป็นค่าระหว่าง 1 ถึง 254

ตัวอย่าง

```
DECLARE A CHARACTER (10);
```

หมายถึง define ให้ตัวแปร A เป็น character string ความยาว 10 ตัว

ถ้า character string ที่ให้เป็นมูลค่าของ A มีความยาวน้อยกว่า A เครื่องจะใส่เครื่องหมาย blanks ทางขวามือของค่านั้น จนกระทั่งมีความยาวเท่ากับ A แต่ถ้า string นั้นมีความยาวมากกว่า A ส่วนที่เกินทางขวามือ ของ string จะถูกตัด

ค่าคงที่ชนิด character string ประกอบด้วย ตัวอักขระต่าง ๆ อยู่ภายในเครื่องหมายคำพูด แต่ถ้าเครื่องหมายคำพูดเป็นส่วนหนึ่งของ string ให้แทนด้วยเครื่องหมายคำพูด 2 ตัวติดกัน ตัวอย่างเช่น ค่าคงที่ string ซึ่งมีค่า เป็น What's Happening? ให้แทนด้วย 'What''s Happening?'

สำหรับ null หรือ empty character string การ define ให้ใช้เครื่องหมายคำพูด 2 ตัวติดกัน, null string จะมีความยาวเท่ากับ 0

ตัวแปรชนิด character string อาจจะมี attribute VARYING ซึ่งหมายความว่า ตัวแปรนั้น อาจจะมีความยาว ได้มากที่สุด n ตัว

ตัวอย่าง

```
DECLARE A CHARACTER(lO) VARYING
```

หมายถึง เตรียมเนื้อที่ A เพื่อเก็บค่าของ character string มีความยาวไม่เกิน 10 ตัว

3.2.2 Bit string data

ตัวแปรที่ declare เป็น BIT(n) หมายถึง ข้อมูลเป็นชนิด bit string ประกอบด้วย เลขฐานสอง n ตัว เมื่อ n มีค่าระหว่าง 1 ถึง 16

ตัวอย่าง

```
DECLARE A BIT(3);
```

หมายถึง เตรียมเนื้อที่เพื่อเก็บ bit string ที่มีความยาวเท่ากับ 3 การกำหนดมูลค่า ให้เก็บตัวแปร ชนิดนี้ ใช้กฎเดียวกับเรื่อง character string ยกเว้น ในกรณีมีการใส่ค่าเพิ่มเนื้อที่นั้น เครื่องจะใส่เลขศูนย์ แทนที่จะเป็น blank

ตัวแปรชนิด bit string จะใช้ attribute VARYING ไม่ได้

ค่าคงที่ bit string จะเป็นชุดของตัวเลข 0-9, ตัวอักษร A-F อยู่ในเครื่องหมายคำพูด และตามด้วย ตัวอักษร B และอาจจะมีเลขอีก 1 ตัว ในช่วง 1 ถึง 4 ตามหลังก็ได้ ซึ่งหมายถึงจำนวน bit ที่ใช้แทนเลขแต่ละตัวในชุดของเลขจำนวนเต็มนั้น นั่นคือ ค่าคงที่ชนิด bit string อาจเขียนเป็นเลขฐานสอง (B หรือ B1 format), ฐานสี่ (B2 format), ฐานแปด (B3 format) หรือ ฐานสิบหก (B4 format) ตัวอักขระหรือตัวเลข ที่อยู่ในชุดของเลขจำนวนนั้น จะต้องเป็นเลขที่ใช้ได้ สำหรับ ฐานเลขที่เรากำหนด โดย format

ตัวอย่าง ค่าคงที่ bit string ที่มีความหมายเหมือนกับรูปแบบฐานสอง

'101'B1 มีความหมายเหมือนกับ '101'B

'101'B2 มีความหมายเหมือนกับ '010001'8

'101'B3 มีความหมายเหมือนกับ '001000001'8

'101'B4 มีความหมายเหมือนกับ '000100000001'B

'9A'B4 มีความหมายเหมือนกับ '10011010'B

'77'B3 มีความหมายเหมือนกับ '111111'8

3.2.3 Concatenation

เราใช้ตัวปฏิบัติการ infix || หรือ !! ในการต่อ bit strings หรือ character strings ทั้งนี้ตัวถูกกระทำ (operands) ทั้งสองต้องเป็นชนิดเดียวกัน ซึ่งก็จะเป็น ชนิดของผลลัพธ์ในตอนท้าย ความยาวของ string ผลลัพธ์ เท่ากับ ความยาวของตัวถูกกระทำ 2 ตัวนั้นบวกกัน สำหรับการต่อ character string ถ้าตัวถูกกระทำตัวใดตัวหนึ่ง มี attribute VARYING ผลลัพธ์ก็จะมี attribute VARYING ด้วย

ตัวอย่าง

```
DECLARE
    A CHARACTER(S),
    B CHARACTER(G),
    C CHARACTER(20) VARYING;
A = 'ABC';
B = 'ABCDPF';
C = A   B
```

ตัวแปร c จะมีค่าเท่ากับ 'ABCABCDEF' ความยาว เท่ากับ 9

3.2.4 String built-in functions

PL/I-80 มี บิลท์-อิน ฟังก์ชัน สำหรับ string ดังนี้ (รายละเอียดเพิ่มเติมให้ดูบทที่ 12)

ASCII	LENGTH
BITS	RANK
BOOL	SUBSTR
CHARACTER	TRANSLATE
COLLATE	UNSPEC
INDEX	VERIFY

3.3 Control data items

control data item เป็น problem data item ใช้สำหรับควบคุมทิศทางของโปรแกรม ตัวอย่างเช่น ไอเดนติไฟเออร์ ที่เราใช้เป็น label ของคำสั่งต่าง ๆ และ label ของ procedure เป็น control data items

3.3.1 Label data

label data ประกอบด้วย ค่าคงที่ label (label statements) และตัวแปร label (label variables), ค่าคงที่ label เป็น label ไอเดนติ-ไฟเออร์ ซึ่งอยู่ข้างหน้าคำสั่งปฏิบัติการ (executable statements) ส่วนตัวแปร label เป็นตัวแปรซึ่งเรา defined ในคำสั่ง DECLARE และมี attribute LABEL กำกับ การกำหนดค่าของค่าคงที่ label หรือ ตัวแปร label อื่น ๆ ให้เป็นไปตามกฎเดียวกับ การกำหนดค่า ให้กับตัวแปรชนิดอื่น ๆ

ทั้งค่าคงที่ label และตัวแปร label ให้ใช้กฎเดียวกับ การกำหนดชื่อ, label data item จะมีความหมายก็ต่อเมื่ออยู่ภายใน block ซึ่งได้ declared explicitly ในคำสั่ง DECLARE หรือได้ถูก declared implicitly โดยตัวมันเอง ให้เป็นค่าคงที่ label นอกจากนี้แล้ว การเคลื่อนย้ายการควบคุม โดยใช้คำสั่ง GOTO จะถูกต้องก็ต่อเมื่อ label เป้าหมาย (target label) มีค่าอยู่ใน block ซึ่งมีคำสั่ง GOTO การเคลื่อนย้ายการควบคุม ด้วยคำสั่ง GOTO และ label ถูกจำกัดเขต เฉพาะใน block ที่กำลังปฏิบัติงานปัจจุบัน หรือ เฉพาะ block ที่มันอยู่

ค่าคงที่ label อาจจะมี subscript โดยใช้ค่าคงที่จำนวนเต็ม 1 ตัว (มีเครื่องหมายกำกับ หรือไม่ก็ได้) สำหรับการเกิดของ label ที่มี subscripted ทั้งหมด กับไอเดนติไฟเออร์ตัวเดียวกัน ใน block เป็นการสร้าง การ declare implicit ของ constant label array สำหรับ block นั้น การเกิดของ ชื่อ label ที่เหมือนกันภายใน block อื่น รวมทั้ง block ที่มันอยู่ เป็นการ define การ declare local ใหม่ให้กับ block นั้น

การ define constant label array ใด ๆ อย่าง implicitly จะมีความหมายก็ต่อเมื่อ ตัว subscripts เหล่านั้น เกิดใน block ที่สมนัยกับมัน ส่วนตัวแปร label อาจจะ defined แบบ explicitly ให้เป็น array ที่มี subscript ตัวเดียว ในคำสั่ง DECLARE

ตัวปฏิบัติการ ซึ่งจะใช้กับ label data มีเพียง 2 ตัวคือ ตัวปฏิบัติการเปรียบเทียบเท่ากับ (equal) และตัวปฏิบัติการไม่เท่ากับ (not equal)

3.3.2 Entry data

entry data items อาจจะเป็นค่าคงที่ entry หรือตัวแปร entry ตัวอย่างเช่น label ของคำสั่ง PROCEDURE เรียกว่า ค่าคงที่เอนทรี (entry constant) ค่าคงที่เอนทรี อาจจะเป็น external (กำหนดตำแหน่งของเอนทรีให้กับ external procedure) หรือ internal (กำหนดตำแหน่งของเอนทรี ให้กับ block ที่ซ้อนกัน)

ตัวแปรซึ่ง declare เป็น ENTRY VARIABLE เป็นตัวแปรเอนทรี ซึ่งอาจจะกำหนดค่า ให้เป็นค่าคงที่เอนทรี หรือ ค่าของตัวแปรเอนทรีอื่น ๆ และเช่นเดียวกับในเรื่อง label data ตัวปฏิบัติการที่จะใช้กับข้อมูลเอนทรี มีเพียง 2 ตัวคือ ตัวปฏิบัติการเปรียบเทียบเท่ากับ และตัวปฏิบัติการไม่เท่ากับ ตัวแปรเอนทรี อาจจะมี subscript ได้

ตัวอย่าง

```
DECLARE
     A  ENTRY  VARIABLE,
     (X,Y)  FLOAT  BINARY,.
     F(3)  ENTRY  (FLOAT)  RETURNS  (FLOAT)  VARIABLE,
     ZZ   ENTRY  (FLOAT)  RETURNS  (FLOAT);
P1:
   PROCEDURE;
      X = 5;
   END;
```

```
P2:
   PROCEDURE;
      X = 25;
   END;
Y = 9;
IF Y = 5 THEN
    A = P1;
ELSE
    A = P2;
CALL A;
F(2) = ZZ;
Y = F(2)(X);
PUT LIST (Y);
```

รายละเอียดเพิ่มเติมเกี่ยวกับ ข้อมูลชนิด entry ให้ดูบทที่ 8

3.4 Pointer data

ข้อมูลชนิด พอยท์เตอร์ ใช้เพื่อบอกตำแหน่งที่อยู่ ภายในหน่วยความจำ ค่าของมันคือ ตำแหน่งที่อยู่ของ ตัวแปรในโปรแกรม การ delare ตัวแปรพอยท์เตอร์ มีรูปแบบดังนี้

```
DECLARE   X  POINTER;
```

ตัวแปรพอยท์เตอร์ อาจจะกำหนดให้เป็นค่าของ ตัวแปรพอยท์เตอร์อื่นได้ และก็เหมือนกับ ในเรื่อง ข้อมูล label และข้อมูลเอนทรี คือตัวปฏิบัติการที่ใช้ได้กับข้อมูลพอยท์เตอร์ มี 2 ตัว ได้แก่ = และ ~=

ชนิดของตัวแปรที่กำหนดไว้ จะต้องใช้ได้กับ ข้อมูลพอยท์เตอร์ เมื่อมีการอ้างถึงมัน รูปแบบ การอ้างถึง pointer-qualified มีดังนี้

pointer-variable -> based-variable

เมื่อ ตัวแปรพอยท์เตอร์ บอกตำแหน่งที่อยู่ ของ based-variable

ตัวอย่าง

```
DECLARE  P  POINTER;

DECLARE  X  CHARACTER(2)  BASED;

     .

P -> X = 'AA';
```

รายละเอียดของข้อมูลพอยท์เตอร์ ให้ดูบทที่ 6

3.5 File data

ข้อมูลชนิด file data ใช้แทน ข้อเท็จจริง (information) เกี่ยวกับ external device ภาษา PL/I-80 ใช้ค่าคงที่แฟ้มข้อมูล (file constant) และตัวแปรแฟ้มข้อมูล (file variable) ในการเข้าถึง (access) external data set

ตัวอย่าง ค่าคงที่แฟ้มข้อมูล

```
DECLARE  fname FILE;
```

เมื่อ fname เป็น ไอเดนติไฟเออร์ กำหนดให้เป็น ชื่อแฟ้มข้อมูล

ในกรณีที่ fname ไม่ใช่พารามิเตอร์ ชื่อแฟ้มข้อมูล จะเป็น EXTERNAL โดยอัตโนมัติ เพื่อว่า มันจะเข้าถึงกลุ่มข้อมูลเดียวกัน ในทุก modules ที่มันถูก declare ได้

หมายเหตุ ถ้าไม่มีการ execute คำสั่ง OPEN ที่มี option TITLE CP/M disk file ชื่อ "fname.DAT" จะถูกเข้าถึงบน default drive

การ declare ตัวแปรแฟ้มข้อมูล ทำดังนี้

```
DECLARE fname FILE VARIABLE;
```

รายละเอียดเพิ่มเติมเกี่ยวกับ file data ให้ดูในบทที่ 9 และในหนังสือคู่มือ "PL/I-80 Applications Guide"