

517 101 Introduction to Computer (Lab 8)

Operator , Expression and Assignment Statement (Part I)*

*เอกสารอ้างอิง : 1. C Programming Language (2nd Edition), Brian W. Kernighan, Dennis M. Ritchie
2. เอกสารประกอบการสอนวิชา 517 111, อาจารย์ โอภาส วงษ์ทวีทรัพย์

5

ขอบเขตของเนื้อหา

- ทบทวนคำสั่งที่ใช้ในการแสดงผลลัพธ์ของข้อมูลออกสู่จอภาพ (*Output Statement via Monitor*) และการประกาศตัวแปร
- ตัวดำเนินการ (*operator*) และ นิพจน์การคำนวณในภาษาโปรแกรม (*Expression*)

5

ทบทวน : โครงสร้างของภาษาซี

```
2- Suhsurfhwvruglhfwyh */  
& hqfoxgh <vwglr.k>
```

```
p dlq() //Ghidxowuhwxuq lqwhjhu  
{ 22Ehjlq rip dlq surjudp  
    ý 22Vwdwhp hqw  
    ý 22Vwdwhp hqw  
    ¢ 22Hqg rip dlq surjudp
```

6

ทบทวน : การแสดงผล printf(“ข้อความ และ/หรือ รูปแบบแสดงผล”);

```
& hqfoxgh ? wvgrlkA
```

```
& ghilqh [ 43
```

```
yrhg p dlq+,
```

```
~
```

```
subqwi+E( g_qS/58,>
```

```
subqwi+E( i_qS/ <1k<<S,>
```

```
subqwi+E[ @ ( g_qS/ [,>
```

```
subqwi+E( f ( fµ ( vS/ 0NÝ 0NÝ EVksdnruq X1S,>
```

```
¢
```

```
25  
9.999  
X = 10  
SU - Silpakorn U.
```

7

บททวน : การประกาศตัวแปร

Syntax:

```
G dwdw|sh Ydubediv>
```

Example:

```
hw frxqw>
```

```
fkdu vh{ @ 3>
```

```
hw d/e /f>
```

```
icrdwvxp @ 3/wrd@ 3>
```

8

ตัวดำเนินการ (operator)

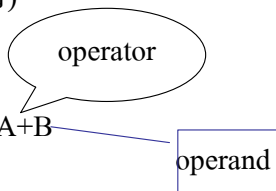
- ตัวดำเนินการ (operator) คือเครื่องหมายที่ใช้เพื่อแสดงความหมายในการกระทำต่างๆ สามารถจัดกลุ่มได้ดังนี้
 1. ตัวดำเนินการเดี่ยว (Unary Operators)
 2. ตัวดำเนินการคู่ (Binary Operators)
 3. ตัวดำเนินการเลขคณิต (arithmetic operators)
 4. ตัวดำเนินการเปรียบเทียบ (relational operator)
 5. ตัวดำเนินการเชิงตรรกะ (logical operator)

9

ตัวถูกดำเนินการ (operand) และ โครงสร้างของนิพจน์ (expression)

- ข้อมูลที่ทำงานกับตัวดำเนินการ เรียกว่า **ตัวถูกดำเนินการ (operand)** ตัวดำเนินการบางตัวต้องมีตัวถูกดำเนินการ 2 ตัว เช่น $A + B$, $A * B$
- ในขณะที่บางตัวจะใช้ตัวถูกดำเนินการเพียงตัวเดียว เช่น $\text{sqrt}(a)$
- นิพจน์ประกอบด้วย 2 ส่วน คือ operand (ตัวถูกดำเนินการ) และ operator (ตัวดำเนินการ)

- เช่น $(A+4)/\text{sqrt}(8)$, $A+B$



:

นิพจน์ (Expression) และ

การเขียนนิพจน์ในภาษาซี

- นิพจน์ หมายถึง การนำตัวแปร ค่าคงที่ มาสัมพันธ์กันโดยใช้เครื่องหมายอย่างหนึ่งอย่างใดเป็นตัวเชื่อม เช่น $(a+b) / (c*d)$
- ห้ามเขียนตัวแปร 2 ตัวติดกันโดยไม่มีเครื่องหมาย เช่น ab ในภาษา C ต้องเขียน $a*b$ จะเขียนเป็น ab ไม่ได้เพราะจะถือเป็นชื่อตัวแปรตัวเดียวชื่อ ab ไม่ใช่ค่า a คูณ b
- ไม่สามารถเขียนตัวห้อย หรือตัวยกกำลังได้
- นิพจน์ที่เราใช้โดยทั่วๆ ไปในสมการทางคณิตศาสตร์จึงไม่สามารถเขียนได้โดยตรง เมื่อเราเขียนโปรแกรม แต่เราจะต้องแปลงนิพจน์เหล่านั้นให้อยู่ในรูปของนิพจน์ในภาษาซีเสียก่อน

;

การเขียนนิพจน์ในภาษาซี

นิพจน์คณิตศาสตร์

$$\frac{a + b}{cd}$$

$$10x + 3xy + 10y$$

นิพจน์ภาษา C

$$(a+b) / (c*d)$$

$$10*x + 3*x*y + 10*y$$

<

ตัวดำเนินการกำหนดค่า (assignment operator)

- ตัวดำเนินการกำหนดค่าที่ใช้กันโดยทั่วไปคือ =

โดยนิพจน์กำหนดค่าที่ใช้ตัวดำเนินการนี้เขียนได้ในรูปแบบดังนี้

```
lghqwlihu@ h{shvvlrq
```

- โดยที่ identifier โดยทั่วไปจะเป็นตัวแปร และ expression จะเป็นค่าคงที่ ตัวแปร หรือนิพจน์ที่มีความซับซ้อนก็ได้
- นิพจน์ทางขวามือ กำหนดค่า ให้กับตัวแปรทางซ้ายมือ

เช่น `x = 50;`

43

ตัวอย่าง นิพจน์กำหนดค่าที่ใช้ตัวดำเนินการ =

```
a = 3
```

```
x = y
```

```
delta = 0.001
```

```
sum = a + b
```

```
area = length * width
```

ในหนึ่งประโยค สามารถใช้ตัวดำเนินการ = ได้หลายตัว

เช่น `lgh qw l/n/frw/ihh>`

`l@ m@ n @ 433>`

`frw@ ihh@ 7-8>`

44

ตัวดำเนินการเดี่ยว (Unary Operators)

Unary Operator is a statement contain one operand and one operand.

+ Unary plus **Ex** `z = +x;`

- Unary minus **Ex** `z = -x;`

++ Increment **Ex** `++x;` `x++;`

-- Decrement **Ex** `--x;` `x--;`

45

ตัวดำเนินการเพิ่ม/ลด

(increment/decrement operator)

- เป็นตัวดำเนินการสำหรับการเพิ่มค่าและการลดค่าในตัวแปรครั้งละหนึ่ง

operator	ชื่อ	การทำงาน
++	increment	เพิ่มค่าขึ้นหนึ่ง
--	decrement	ลดค่าลงหนึ่ง

- ซึ่งสามารถนำไปใช้กับตัวแปรได้ 2 ลักษณะ คือ

46

1. วางไว้หลังตัวแปร เช่น `i++` หรือ `i--`

- ซึ่งในกรณีนี้จะทำ operator ตัวอื่นก่อน แล้วจึงจะเพิ่มหรือลดค่าในตัวแปร

เช่น `count = 2;`

`i = count++;`

ดังนั้นจะนำค่าในตัวแปร `count` กำหนดให้กับ `i` ก่อน แล้วจึงเพิ่มค่าในตัวแปร `count` ขึ้นอีกหนึ่ง

จะได้ `i` มีค่าเท่ากับ 2

`count` มีค่าเท่ากับ 3

47

2. วางไว้หน้าตัวแปร เช่น `++i` หรือ `--i`

- ซึ่งในกรณีนี้จะเพิ่มหรือลดค่าในตัวแปรก่อน แล้วจึงทำ operator ตัวอื่น

เช่น `count = 2;`

`i = ++count;`

ดังนั้นจะเพิ่มค่าในตัวแปร `count` ขึ้นอีกหนึ่ง ก่อนจะกำหนดให้กับตัวแปร `i`

จะได้ `i` มีค่าเท่ากับ 3

`count` มีค่าเท่ากับ 3

48

Compound assignments

- ในประโยคที่มีการกำหนดค่าใหม่ให้กับตัวแปร โดยใช้ค่าเดิมในตัวแปรไปทำการคำนวณ เช่น

`int i = 5;`

`i = i + 5; /* i เท่ากับ 10 */`

คือ จะมีการนำค่าเดิมในตัวแปร `i` บวกด้วย 5 แล้วนำผลลัพธ์ที่ได้กำหนดให้ตัวแปร `i` อีกครั้ง

- ในการทำงานลักษณะนี้ C ได้จัดเตรียม operator ในกลุ่มที่เรียกว่าที่เรียกว่า **compound assignments** เพื่อให้ผู้ใช้สามารถเรียกใช้ได้ง่ายขึ้น operator ในกลุ่มของ compound assignment ประกอบด้วย

49

ตารางตัวอย่างการใช้ compound assignments

Operator	ตัวอย่าง	เทียบเท่ากับ
+=	i += 5;	i = i + 5;
-=	i -= 5;	i = i - 5;
*=	i *= 5;	i = i * 5;
/=	i /= 5;	i = i / 5;
%=	i %= 5;	i = i % 5;

4:

ตัวดำเนินการคู่ (Binary Operators)

Binary Operator is a statement contain one operator and two operands ex $Z = X + Y$;

- * Multiply / Divide
- % Remainder (modulus)
- + Binary plus - Binary minus

Ex $Z = X * Y$; $A = B/2 - 6$; $F = (D + 7) - C$;

4:

ตัวดำเนินการเลขคณิต (arithmetic operators)

เป็นตัวดำเนินการที่ทำให้เกิดการกระทำทางคณิตศาสตร์ ประกอบด้วย

operator	ความหมาย
+	การบวก
-	การลบ และ unary operator
*	การคูณ
/	การหาร ซึ่งจะเป็นการหาผลลัพธ์จากการหาร
% (modulo)	การหาเศษ(remainder) ที่เป็นจำนวนเต็มจากการหาร (operand ต้องเป็นจำนวนเต็มทั้งคู่)

กำหนดลำดับความสำคัญของตัวดำเนินการ

(Precedences of Operators)

- จะเกิดอะไรขึ้น กับการเขียนนิพจน์ในรูปแบบนี้
 $result = a + b * c * d / 5 - 7 + f * g$;
- คอมพิวเตอร์จะมีวิธีการคำนวณอย่างไร???
- ดังนั้นเราจึงจำเป็นต้องกำหนดลำดับความสำคัญของตัวดำเนินการ ว่าตัวดำเนินการแต่ละตัวนั้น ตัวใดจะถูกกระทำก่อนและหลัง เรียกว่า *Precedence of Operators*

53

กำหนดลำดับความสำคัญของตัวดำเนินการ

(Precedences of Operators)

- โดยมีลำดับความสำคัญที่เราใช้กันบ่อยๆ คือ
 1. ถ้ามีวงเล็บต้องกระทำภายในวงเล็บก่อน
 2. ตัวดำเนินการ * และ /
 3. ตัวดำเนินการ + และ -

54

Associativity

- $result = A * B + C / D;$
- เราจะเริ่มทำจากไหนนะ เห็นบอกว่า ต้องทำ * และ / ก่อน แล้ว * กับ / มีลำดับความสำคัญเท่ากัน จะทำตัวไหนก่อน???
- ในภาษาโปรแกรมจะมีการกำหนดว่า ถ้ามี **Operators** ที่มีลำดับความสำคัญเท่ากันจำนวนมากกว่า 1 ตัวขึ้นไป จะมีการกำหนดทิศทางของการดำเนินการ
- แบ่งเป็น 2 แบบ คือ *ทำจากซ้ายไปขวา* *ทำจากขวาไปซ้าย*

55

ตัวอย่าง แสดงขั้นตอนการทำงานของตัวดำเนินการ

■	$4 + 3 * 2$ $4 + 6$
■	$9 * 2 - 15/3 + 7$ $18 - 5 + 7$ $13 + 7$ 20

56

สรุป ลำดับความสำคัญของ operator

ลำดับความสำคัญมาก	กลุ่มของตัวดำเนินการ	ตัวดำเนินการ	
↓	ตัวดำเนินการยูนิารี	- ++ -- ! sizeof (type)	
	คูณหารและหาเศษเหลือ	* / %	
	บวกและลบ	+ -	
	ตัวดำเนินการเปรียบเทียบ	< <= > >=	
	ตัวดำเนินการเทียบเท่า	= !=	
	AND	&&	
	OR		
	ตัวดำเนินการเงื่อนไข	? :	
	ลำดับความสำคัญน้อย	ตัวดำเนินการกำหนดค่า	= += -= *= /= %=

57