

บทที่ 4 ตัวแปร (Variables)

รายวิชา สธ 113 การออกแบบโปรแกรมทางธุรกิจเบื้องต้น

อ.อภิพงศ์ ปิงยศ

การประกาศตัวแปร (Variable Declaration)

- ▶ ตัวแปรในภาษาซีแบ่งได้ 2 ประเภทใหญ่ๆ คือ ตัวแปรพื้นฐานที่หมายถึงตัวแปรที่เก็บข้อมูลได้เพียงค่าเดียว และตัวแปรชุด คือ ตัวแปรที่สามารถเก็บข้อมูลไว้ได้หลายค่าภายในตัวแปรตัว
- ▶ ตัวแปร (Variable) คือ การจองพื้นที่ในหน่วยความจำของคอมพิวเตอร์สำหรับเก็บข้อมูลที่ต้องใช้ในการทำงานของโปรแกรม โดยมีการตั้งชื่อเรียกหน่วยความจำในตำแหน่งนั้นด้วย เพื่อความสะดวกในการเรียกใช้ข้อมูล ถ้าจะใช้ข้อมูลใดก็ให้เรียกผ่านชื่อของตัวแปรที่เก็บเอาไว้

ชนิดของข้อมูลในภาษาซี (Data Type)

- ▶ 1) ข้อมูลชนิดตัวเลขจำนวนเต็ม (Integer) คือข้อมูลที่เป็นเลขจำนวนเต็ม ได้แก่ จำนวนเต็มบวก จำนวนเต็มลบ ศูนย์ ใช้พื้นที่ในการเก็บ 2 ไบต์
- ▶ 2) ข้อมูลชนิดตัวเลขทศนิยม (Float) คือข้อมูลที่เป็นเลขทศนิยม ขนาด 4 ไบต์
- ▶ 3) ข้อมูลชนิดเลขฐานแปด (Octal)
- ▶ 4) ข้อมูลชนิดเลขฐานสิบหก (Hexadecimal)
- ▶ 5) ข้อมูลชนิดตัวอักขระ (Character) คือข้อมูลที่เป็นรหัสแทนตัวอักษรหรือค่าจำนวนเต็มได้แก่ ตัวอักษร ตัวเลข และกลุ่มตัวอักขระพิเศษใช้พื้นที่ในการเก็บข้อมูล 1 ไบต์
- ▶ 6) ข้อมูลชนิดข้อความ (String)

ตัวแปรพื้นฐานในภาษาซีที่กำหนดตามมาตรฐาน ANSI C

ชนิดของตัวแปร	ความหมาย
char	เก็บข้อมูลชนิดอักขระ โดยใช้พื้นที่ในการจัดเก็บ 8 บิต
unsigned char	เก็บข้อมูลชนิดอักขระแบบไม่คิดเครื่องหมาย
int	เก็บข้อมูลชนิดตัวเลขจำนวนเต็ม โดยใช้พื้นที่ในการจัดเก็บ 16 บิต
unsigned int	เก็บข้อมูลชนิดตัวเลขจำนวนเต็มแบบไม่คิดเครื่องหมาย
short	เก็บข้อมูลชนิดตัวเลขจำนวนเต็มแบบสั้น โดยใช้พื้นที่ในการจัดเก็บ 8 บิต
unsigned short	เก็บข้อมูลชนิดตัวเลขจำนวนเต็มแบบสั้น โดยไม่คิดเครื่องหมาย
long	เก็บข้อมูลชนิดตัวเลขจำนวนเต็มแบบยาว โดยใช้พื้นที่ในการจัดเก็บ 32 บิต
unsigned long	เก็บข้อมูลชนิดตัวเลขจำนวนเต็มแบบยาว โดยไม่คิดเครื่องหมาย
float	เก็บข้อมูลชนิดตัวเลขทศนิยม โดยใช้พื้นที่ในการจัดเก็บ 32 บิต
double	เก็บข้อมูลชนิดตัวเลขทศนิยม โดยใช้พื้นที่ในการจัดเก็บ 64 บิต
long double	เก็บข้อมูลชนิดตัวเลขทศนิยม โดยใช้พื้นที่ในการจัดเก็บ 128 บิต

รูปแบบการประกาศตัวแปร

- ▶ ในภาษาซีมีรูปแบบการประกาศตัวแปร ดังนี้

`type variable;`

type คือ ชนิดของตัวแปรที่จะสร้าง
variable คือ ชื่อของตัวแปรที่ต้องการใช้

- ▶ ตัวอย่างการประกาศตัวแปร เช่น `int num;`
`float grade;`
- ▶ หากต้องการประกาศตัวแปรหลายตัวชนิดเดียวกันก็สามารถทำได้ โดยการใช้เครื่องหมาย ,
คั่น เช่น `int a,b,c;`
- ▶ การประกาศตัวแปร หากอยู่ก่อนฟังก์ชันใดๆ จะเป็น Global Variables แต่หากอยู่ภายใน
ฟังก์ชันใด หลังจากเครื่องหมาย { จะเป็น Local Variable

รูปแบบการประกาศตัวแปร [cont.]

- ▶ นอกจากนี้ยังสามารถกำหนดค่าเริ่มต้นให้กับตัวแปรไปพร้อมกับการประกาศตัวแปร เช่น

```
int num = 1;
```

```
char ch = '#', d = 'D';
```

- ▶ โดยหลักการตั้งชื่อตัวแปรมาใช้งานนั้น ควรคำนึงถึงว่าจะต้องตั้งให้ถูกต้องตามข้อกำหนดของภาษาซี และ ควรจะตั้งชื่อตัวแปรให้สอดคล้องกับการทำงานหรือหน้าที่ของตัวแปรนั้นๆ เพราะเมื่อถึงเวลาต้องมาทำการปรับปรุงแก้ไขโปรแกรม จะสามารถทำได้โดยไม่ยากนัก
- ▶ ชื่อตัวแปร ประกอบด้วย ตัวอักษร ตัวเลข เครื่องหมาย_ และต้องไม่ขึ้นต้นด้วยตัวเลข ตัวแปรมีความยาวเท่าไรก็ได้ แต่คอมไพเลอร์ตามมาตรฐาน ANSI C จะพิจารณาเฉพาะ 31 ตัวแรกเท่านั้น
- ▶ ชื่อ A1 และ a1 จะเป็นคนละตัวแปร
- ▶ ห้ามตั้งชื่อซ้ำกับคำสงวน (Reserved Word)

คำสงวน (Reserved Word)

auto	default	float	register	struct	volatile	break
do	far	return	switch	while	case	double
goto	short	typedef	char	else	if	signed
union	const	enum	int	sizeof	unsigned	continue
extern	long	static	void			

ตัวอย่างการตั้งชื่อ

bath_room	ถูกต้อง
n-sync	ผิดพลาดการ เนื่องจากมีเครื่องหมาย - ปรากฏในชื่อ
108dots	ผิดพลาดการ เนื่องจากขึ้นต้นด้วยตัวเลข
Year#	ผิดพลาดการ เนื่องจากมีเครื่องหมาย # อยู่ในชื่อ
_good	ถูกต้อง
goto	ผิดพลาดการ เนื่องจากเป็นคำสงวน
work	ถูกต้อง
break	ผิดพลาดการ เนื่องจากเป็นคำสงวน

การรับ/แสดงข้อมูล

- ▶ ในการเขียนโปรแกรมนั้น เราจะต้องมีการใช้ Place Holders ในการรับข้อมูลจากผู้ใช้ (ผ่านคำสั่ง scanf) หรือแสดงข้อมูลต่อผู้ใช้(ผ่านคำสั่ง printf) โดยรูปแบบการใช้ Place Holders นั้น จะขึ้นอยู่กับชนิดของตัวแปร เช่น

```
#include <stdio.h>

void main()
{
    int product_price;
    printf("How much is that ? \n");
    scanf("%d", &product_price);
    printf("oh! %d ?, it's too cheap\n\n", product_price);
}
```

Control Character

Control Character	Explanation
<code>%c</code>	a single character
<code>%d</code>	a decimal integer
<code>%i</code>	an integer
<code>%e</code>	scientific notation, with a lowercase "e"
<code>%E</code>	scientific notation, with a uppercase "E"
<code>%f</code>	a floating-point number
<code>%g</code>	use <code>%e</code> or <code>%f</code> , whichever is shorter
<code>%G</code>	use <code>%E</code> or <code>%f</code> , whichever is shorter
<code>%o</code>	an octal number
<code>%x</code>	unsigned hexadecimal, with lowercase letters
<code>%X</code>	unsigned hexadecimal, with uppercase letters
<code>%u</code>	an unsigned integer
<code>%s</code>	a string
<code>%x</code>	a hexadecimal number
<code>%p</code>	a pointer
<code>%n</code>	the argument shall be a pointer to an integer into which is placed the number of characters written so far
<code>%%</code>	a percent sign

```
1  /* Fig. 2.4: fig02_04.c
2     Addition program */
3  #include <stdio.h>
4
5  /* function main begins program execution */
6  int main()
7  {
8     int integer1; /* first number to be input by user */
9     int integer2; /* second number to be input by user */
10    int sum;      /* variable in which sum will be stored */
11
12    printf( "Enter first integer\n" ); /* prompt */
13    scanf( "%d", &integer1 );        /* read an integer */
14
15    printf( "Enter second integer\n" ); /* prompt */
16    scanf( "%d", &integer2 );        /* read an integer */
17
18    sum = integer1 + integer2;        /* assign total to sum */
19
20    printf( "Sum is %d\n", sum );    /* print sum */
21
22    return 0; /* indicate that program ended successfully */
23
24 } /* end function main */
```

Program Output

```
Enter first integer
45
Enter second integer
72
Sum is 117
```

การคำนวณทางคณิตศาสตร์

สัญลักษณ์ที่ใช้ในการคำนวณ	ความหมาย	ตัวอย่าง
+	การบวก	$5.0 + 2.0 = 7.0$
-	การลบ	$5.0 - 2.0 = 3.0$
*	การคูณ	$5.0 * 2.0 = 10.0$
/	การหาร	$5.0 / 2.0 = 2.5$
%	การหาค่าเศษ	$5 \% 2 = 1$

การคำนวณทางคณิตศาสตร์ [cont.]

เครื่องหมาย	ตัวอย่างการใช้งาน	ความหมาย
++	$y = ++X$ $y = X++$	บวกค่าในตัวแปร X เพิ่มขึ้น 1 ก่อนที่จะกำหนดค่า X ให้กับตัวแปร y กำหนดค่า X ให้กับตัวแปร y ก่อนที่จะบวกค่า X เพิ่มขึ้น 1
--	$y = --X$ $y = X--$	ลบค่าในตัวแปร X ลง 1 ก่อนที่จะกำหนดค่า X ให้กับตัวแปร y กำหนดค่า X ให้กับตัวแปร y ก่อนที่จะลดค่า X ลง 1
+=	$y += X$	บวกค่าในตัวแปร y ด้วยค่าในตัวแปร X ผลลัพธ์ที่ได้กำหนดกลับไปให้ y
-=	$y -= X$	ลบค่าในตัวแปร y ด้วยค่าในตัวแปร X ผลลัพธ์ที่ได้กำหนดกลับไปให้ y
*=	$y *= X$	คูณค่าในตัวแปร y ด้วยค่าในตัวแปร X ผลลัพธ์ที่ได้กำหนดกลับไปให้ y
/=	$y /= X$	หารค่าในตัวแปร y ด้วยค่าในตัวแปร X ผลลัพธ์ที่ได้กำหนดกลับไปให้ y
%=	$y %= X$	หารค่าในตัวแปร y ด้วยค่าในตัวแปร X เศษจากการหารเป็นผลลัพธ์ที่ได้ กำหนดกลับไปให้ y

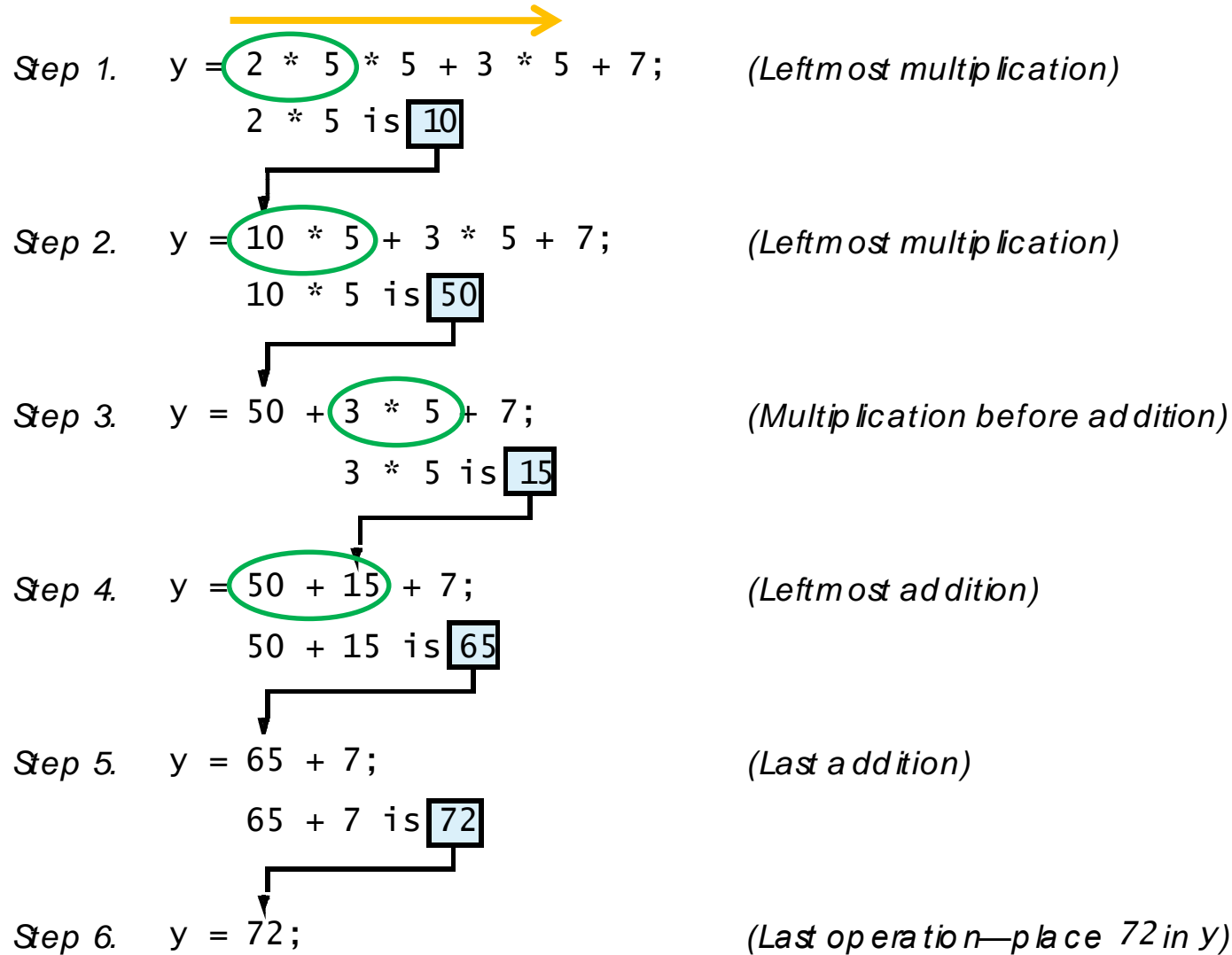
เครื่องหมายที่ใช้ในการเปรียบเทียบ

เครื่องหมาย	การเปรียบเทียบ	ตัวอย่างการใช้งาน	ความหมาย
$==$	เท่ากับ	$x == y$	ผลลัพธ์จะเป็นจริง ถ้าค่าในตัวแปร X เท่ากับค่าในตัวแปร y
$!=$	ไม่เท่ากับ	$x != y$	ผลลัพธ์จะเป็นจริง ถ้าค่าในตัวแปร X ไม่เท่ากับค่าในตัวแปร y
$<$	น้อยกว่า	$x < y$	ผลลัพธ์จะเป็นจริง ถ้าค่าในตัวแปร X น้อยกว่าค่าในตัวแปร y
$<=$	น้อยกว่าหรือเท่ากับ	$x <= y$	ผลลัพธ์จะเป็นจริง ถ้าค่าในตัวแปร X น้อยกว่า หรือ เท่ากับค่าในตัวแปร y
$>$	มากกว่า	$x > y$	ผลลัพธ์จะเป็นจริง ถ้าค่าในตัวแปร X มากกว่าค่าในตัวแปร y
$>=$	มากกว่าหรือเท่ากับ	$x >= y$	ผลลัพธ์จะเป็นจริง ถ้าค่าในตัวแปร X มากกว่า หรือเท่ากับค่าในตัวแปร y

ลำดับความสำคัญในการทำงาน

ลำดับความสำคัญ	เครื่องหมาย	การดำเนินการ
1	()	Left to Right
2	!, ++, --, (typecast)	<i>Right to Left</i>
3	*, /, %	Left to Right
4	+, -	Left to Right
5	<, <=, >, >=	Left to Right
6	==, !=	Left to Right
7	&&	Left to Right
8		Left to Right
9	*=, /=, %=, +=, -=	Left to Right

ตัวอย่างลำดับการทำงานของ $y = 2 * 5 * 5 + 3 * 5 + 7$



```
1  /* Fig. 2.9: fig02_09.c
2     Using if statements, relational
3     operators, and equality operators */
4  #include <stdio.h>
5
6  /* function main begins program execution */
7  int main()
8  {
9     int num1, /* first number to be read from user */
10    int num2; /* second number to be read from user */
11
12    printf( "Enter two integers, and I will tell you\n" );
13    printf( "the relationships they satisfy: " );
14
15    scanf( "%d%d", &num1, &num2 ); /* read two integers */
16
17    if ( num1 == num2 ) {
18        printf( "%d is equal to %d\n", num1, num2 );
19    } /* end if */
20
21    if ( num1 != num2 ) {
22        printf( "%d is not equal to %d\n", num1, num2 );
23    } /* end if */
24
```

fig02_09.c (Part 1
of 2)

```
25     if ( num1 < num2 ) {
26         printf( "%d is less than %d\n", num1, num2 );
27     } /* end if */
28
29     if ( num1 > num2 ) {
30         printf( "%d is greater than %d\n", num1, num2 );
31     } /* end if */
32
33     if ( num1 <= num2 ) {
34         printf( "%d is less than or equal to %d\n", num1, num2 );
35     } /* end if */
36
37     if ( num1 >= num2 ) {
38         printf( "%d is greater than or equal to %d\n", num1, num2 );
39     } /* end if */
40
41     return 0;    /* indicate that program ended successfully */
42
43 } /* end function main */
```

fig02_09.c (Part 2 of 2)

```
Enter two integers, and I will tell you
the relationships they satisfy: 3 7
3 is not equal to 7
3 is less than 7
3 is less than or equal to 7
```

Program Output

Program Output (continued)

```
Enter two integers, and I will tell you
the relationships they satisfy: 22 12
22 is not equal to 12
22 is greater than 12
22 is greater than or equal to 12
```

```
Enter two integers, and I will tell you
the relationships they satisfy: 7 7
7 is equal to 7
7 is less than or equal to 7
7 is greater than or equal to 7
```

References

- ▶ เอกสารประกอบการสอนรายวิชา 259201 Computer Programming for Engineers ปีการศึกษา 2556, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
- ▶ ธีรวัฒน์ ประกอบผล, คู่มือการเขียนโปรแกรมภาษา C ฉบับสมบูรณ์ (พิมพ์ครั้งที่ 4). กรุงเทพฯ : รีไควว่า, 2557