




# </> ตัวแปร ตัวดำเนินการ การนำข้อมูลเข้า และการแสดงผล

ค่าย 1 คอมพิวเตอร์โอลิมปิก ศูนย์ สอวน.มหาวิทยาลัยวลัยลักษณ์  
วันที่ 25 มีนาคม 2565



# ตัวดำเนินการ

ตัวดำเนินการ(Operator) ในการเขียนโปรแกรมภาษาซี มีอยู่ 3 ประเภท  
ใหญ่ ๆ ดังนี้

- 1) ตัวดำเนินการทางคณิตศาสตร์ (arithmetic operators)
- 2) ตัวดำเนินการทางตรรกะ (logical operator)
- 3) ตัวดำเนินการเปรียบเทียบ (relational operator)

# ตัวดำเนินการทางคณิตศาสตร์ (arithmetic operators)

เป็นตัวดำเนินการทางคณิตศาสตร์ ได้แก่ เครื่องหมายที่ใช้ในการบวก ลบ คูณ หาร ตัวเลข และอื่น ๆ ดังตาราง

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
+	การบวก	$X + Y$
-	การลบ	$X - Y$
*	การคูณ	$X * Y$
/	การหาร	$X / Y$
%	การหารจะเอาเฉพาะเศษไว้	$11\%3=(3 \text{ เศษ } 2)$ ดังนั้น 2 เป็นผลลัพธ์
--	การลดค่าลงครั้งละ 1	$X--$ หรือ $--X$ เหมือนกับ $X=X-1$
++	การเพิ่มค่าครั้งละ 1	$X++$ หรือ $++X$ เหมือนกับ $X=X+1$

# ขั้นตอนการทำงานของตัวดำเนินการในภาษาซี

ลำดับความสำคัญ  
ของตัวดำเนินการ

ตัวดำเนินการ	ความสำคัญ
( )	1 (สูงสุด)
++ -- unary	2
* / %	3
+ -	4
+= -= *= /= %=	5 (ต่ำสุด)

ตัวอย่าง

$5 + 6 * 3 = 23$	ตัวดำเนินการ * อยู่ลำดับสูงกว่า + จึงต้องคูณเลขก่อนแล้วบวก 5 ภายหลัง
$2*3-14/7+5 = 9$	1. ตัวดำเนินการ * และ / อยู่ลำดับเดียวกัน ให้ทำจาก ซ้ายไปขวา คือคำนวณคูณก่อนแล้วหาร 2. ตัวดำเนินการ - และ + อยู่ลำดับเดียวกัน ทำจากซ้ายมือก่อน คือ ลบเลข แล้วจึงบวกเลขในภายหลัง

# ตัวดำเนินการเอกภาค (unary operator)

เป็นการใช้ตัวดำเนินการกับตัวแปรตัวเดียว มีการใช้สองแบบคือ

## 1. ตัวดำเนินการเอกภาคเติมหลัง (postfix mode)

EX: `a++`;      เพิ่มค่าครั้งละหนึ่งค่า

## 2. ตัวดำเนินการเอกภาคเติมหน้า (prefix mode)

EX: `++a`;      เพิ่มค่าครั้งละหนึ่งค่า

นำค่า `x` ให้ตัวแปร `y` ก่อน  
จากนั้นเพิ่มค่า `x` ขึ้นหนึ่ง

`y = x++`;

`y = ++x`;

เพิ่มค่า `x` ขึ้นหนึ่งแล้ว  
ใส่ให้ตัวแปร `y`

การคำนวณ	ตัวดำเนินการ	ตัวอย่าง	การทำงาน
เพิ่มค่าทีละหนึ่ง	++	<code>x++</code>	<code>x = x + 1</code>
		<code>++x</code>	<code>x = x + 1</code>
ลดค่าทีละหนึ่ง	--	<code>x--</code>	<code>x = x - 1</code>
		<code>--x</code>	<code>x = x - 1</code>

# ตัวดำเนินการประกอบ

เป็นการใช้ตัวดำเนินการหนึ่งตัวร่วมกับเครื่องหมายเท่ากับ ซึ่งจะเป็นตัวดำเนินการแบบลัดรูป

ตัวดำเนินการประกอบ	ตัวอย่าง	การทำงาน
+=	<code>x += 5;</code>	<code>x = x + 5</code>
-=	<code>x -= 5;</code>	<code>x = x - 5</code>
*=	<code>x *= y;</code>	<code>x = x * y</code>
/=	<code>x /= y;</code>	<code>x = x / y</code>
%=	<code>x %= 5;</code>	<code>x = x % 5</code>
+=	<code>x += y/8;</code>	<code>x = x + y / 8</code>

## ตัวอย่างการใช้งาน

คำสั่ง	x	y
<code>int x = 10, y = 20;</code>	10	20
<code>++x;</code>	11	20
<code>y = - -x;</code>	10	10
<code>x = x-- +y;</code>	19	10
<code>y = x- ++x;</code>	20	0

# ตัวคำนวณทางตรรกะ (logical operators)

เครื่องหมายตรรกะ มีจุดประสงค์ใช้ในการเปรียบเทียบเพื่อตัดสินใจ

เครื่องหมาย	ความหมาย	ตัวอย่าง
&&	และ	$x < 60 \ \&\& \ x > 50$ กำหนดให้ $x$ มีค่าในช่วง 50 ถึง 60
	หรือ	$x == 10 \    \ x == 15$ กำหนดให้ $x$ มีค่าเท่ากับตัวเลข 2 ค่าคือ 10 หรือ 15
!	ไม่	$x = 10 \ !x$ กำหนดให้ $x$ ไม่เท่ากับ 10

# ตัวคำนวณเปรียบเทียบ (relational operators)

เครื่องหมาย	ความหมาย	ตัวอย่าง
>	มากกว่า	$a > b$ $a$ มากกว่า $b$
>=	มากกว่า หรือ เท่ากับ	$a >= b$ $a$ มากกว่าหรือเท่ากับ $b$
<	น้อยกว่า	$a < b$ $a$ น้อยกว่า $b$
<=	น้อยกว่า หรือ เท่ากับ	$a <= b$ $a$ น้อยกว่าหรือเท่ากับ $b$
==	เท่ากับ	$a == b$ $a$ เท่ากับ $b$
!=	ไม่เท่ากับ	$a != b$ $a$ ไม่เท่ากับ $b$

# นิพจน์

นิพจน์(Expression) คือ การนำตัวแปร ค่าคงที่ และตัวดำเนินการมาเขียนประกอบกัน โดยใช้เครื่องหมายอย่างใดอย่างหนึ่งเป็นตัวเชื่อม และเพื่อให้ตัวแปรภาษาสามารถเข้าใจและคำนวณหาผลลัพธ์ได้ตามที่เราต้องการ

- 1) เขียนตัวอักษรหลายตัวติดกันได้โดยไม่มีเครื่องหมายคั่น เช่น XY ถือเป็น 1 ตัวแปร
- 2) กรณีนิพจน์มีค่าของตัวแปรหรือค่าคงที่ต่างชนิดกันในนิพจน์เดียวกัน กลไกของภาษาซี จะเปลี่ยนชนิดของข้อมูลที่มีขนาดเล็กให้เป็นชนิดของข้อมูลที่ใหญ่ขึ้น

นิพจน์ทางคณิตศาสตร์ตามปกติ	นิพจน์ทางคณิตศาสตร์ในภาษาซี
$x + y - z$	$x + y - z$
$2xy + 4z$	$2 * x * y + 4 * z$
$x^2 + 2x + 1$	$x * x + 2 * x + 1$
$2r$	$2 * r$
$\frac{a - b}{c + d}$	$(a - b) / (c + d)$



# การนำข้อมูลเข้า

## ฟังก์ชันรับข้อมูล (Input Function)

<code>scanf( )</code>	รับข้อมูลจากคีย์บอร์ดมาเก็บในตัวแปรที่กำหนด
<code>getchar( )</code>	รับข้อมูลอักขระตัวเดียว และต้องกด Enter ทุกครั้ง
<code>getch( )</code>	รับข้อมูลอักขระตัวเดียว ไม่ต้องกด Enter และไม่แสดงผล
<code>getche( )</code>	รับข้อมูลอักขระตัวเดียวเหมือน getch( ) แต่จะแสดงผลด้วย
<code>gets( )</code>	รับข้อความ (string) มาเก็บในตัวแปรที่กำหนด

# scanf( )

`scanf( “control string”, argument list);`

control string เป็นรหัสรูปแบบข้อมูล (format code)

argument list ตัวแปรที่ใช้เก็บข้อมูล ใช้เครื่องหมาย & นำหน้าชื่อ

แต่ถ้ารับข้อมูลเป็นสตริงไม่ต้องมีเครื่องหมายนี้

# format code

รหัสรูปแบบ	ชนิดตัวแปร	ลักษณะการแสดงผลออกจอภาพ
%d	int	ใช้แสดงข้อมูลที่เป็นเลขจำนวนเต็มฐานสิบ
%ld	long int	ใช้แสดงข้อมูลที่เป็นเลขจำนวนเต็มฐานสิบแบบ long
%u	unsigned int	ใช้แสดงข้อมูลที่เป็นเลขจำนวนเต็มฐานสิบแบบ unsigned
%c	char	ใช้แสดงข้อมูลที่เป็นตัวอักษร
%s	string	ใช้แสดงข้อมูลที่เป็นตัวแปรสตริงหรือชุดตัวอักษร
%o	int (octal)	ใช้แสดงข้อมูลที่เป็นเลขฐานแปด
%x	int (hexa)	ใช้แสดงข้อมูลที่เป็นเลขฐานสิบหก
%f	float	ใช้แสดงข้อมูลที่เป็นเลขทศนิยม ที่ไม่มีเลขยกกำลัง
%e	float, double	ใช้แสดงข้อมูลที่เป็นเลขทศนิยม ที่มีเลขชี้กำลัง
%lf	double	ใช้แสดงข้อมูลที่เป็นเลขทศนิยมแบบ double

Int a;

scanf(“%d”,&a);

char name;

scanf(“%c”,&name);

# getchar( )

```
getchar ();
```

หรือ 

```
char_var = getchar();
```

char\_var เป็นตัวแปรชนิด char ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่านคีย์บอร์ด

```
char d;
```

```
d = getchar();
```

# getch()

```
getch();
```

หรือ 

```
char_var = getch();
```

char\_var เป็นตัวแปรชนิด char ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่านคีย์บอร์ด ฟังก์ชันนี้ใช้รับอักขระตัวเดียวจากคีย์บอร์ด ไม่ต้องกด Enter ข้อมูลที่รับเข้าไปจะไม่แสดงผล

```
char d;
```

```
d = getch();
```

# getche()

```
getche();
```

หรือ `char_var = getche();`

`char_var` เป็นตัวแปรชนิด `char` ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่านคีย์บอร์ด ฟังก์ชันนี้  
ใช้รับอักขระตัวเดียวจากคีย์บอร์ด ไม่ต้องกด Enter ข้อมูลที่รับเข้าไปจะแสดงผล

```
char d;
```

```
d = getche();
```

# gets( )

```
gets(string_var);
```

string\_var เป็นตัวแปรสตริงที่ใช้เก็บข้อความ (string constant)  
ฟังก์ชันนี้ใช้รับข้อมูลจากคีย์บอร์ดมาเก็บในตัวแปรสตริง

```
char d[30];  
gets(d);
```

# การแสดงผล

## ฟังก์ชันแสดงข้อมูล (Output Function)

**printf ()** แสดงผลลัพธ์ออกทางจอภาพตามรูปแบบที่กำหนด

**putchar()** แสดงตัวอักขระตัวเดียวออกทางจอภาพ

**puts()** แสดงข้อความสตริงออกทางจอภาพ



# printf( )

**printf( “control”, value);**

control : เป็นส่วนควบคุมการแสดงผล ประกอบด้วยข้อความ รหัสควบคุม (เช่น %d %f) และอักขระควบคุมการแสดงผล(\n)

value : ได้แก่ค่าตัวแปร นิพจน์ ที่ต้องการแสดงผล ถ้าหากมีมากกว่าหนึ่งตัว ให้ใช้ comma คั่นระหว่างแต่ละตัว

รหัสควบคุมการแสดงผล (escape character)	ใช้สำหรับ
\n	ขึ้นบรรทัดใหม่ 1 บรรทัด
\t	tab เว้นวรรค 1 tab (8 อักขระ)
\a	ส่งเสียงบี๊บเตือน 1 ครั้ง
\'	แสดงอักขระ ' 1 ตัว
\"	แสดงอักขระ " 1 ตัว
\\	แสดงอักขระ \ 1 ตัว

```
printf(“%d“,x1);
```

```
printf(“%x %o “,x2,x3);
```

```
printf(“%f %e “,y1,y2);
```

```
printf(“%c %s “,z,name);
```

# Program 1

```
#include<stdio.h>
void main(void)
{
    printf("Hello world my name is ..... ");
}
```



Hello world my name is .....

# Program 2

```
#include <stdio.h>
#include <conio.h>
void main( )
{
    clrscr( );
    printf("\n[%s]", "Computer");           /* [Computer] */
    printf("\n[%2s]", "Computer");         /* [Computer] */
    printf("\n[%.3s]", "Computer");        /* [Com] */
    printf("\n[%10s]", "Computer");        /* [ Computer] */
    printf("\n[%-10s]", "Computer");       /* [Computer ] */
    printf("\n[%-10.3s]", "Computer");     /* [Com ] */
}
```

# Program 3

```
#include <stdio.h>
#include <conio.h>

void main( )
{
    printf("\n[%x]", 15);                /* Hexa [f] */
    printf("\n[%o]", 15);                /* Octal [17] */
    printf("\n"); printf("\n[%d]", 100); /* [100] */
    printf("\n[%2d]", 100);               /* [100] */
    printf("\n[%10d]", 100);              /* [ 100] */
    printf("\n[%-10d]", 100);             /* [100 ] */
    printf("\n[%-10.2d]", 100);           /* [100 ] */
    printf("\n");
    printf("\n[%f]", 32.5762);            /* [32.576200] */
    printf("\n[%2f]", 32.5762);           /* [32.58] */
    printf("\n[%10.2f]", 32.5762);        /* [ 32.58] */
    printf("\n[%-10.2f]", 32.5762);      /* [32.58 ] */
}
```

# putchar() และ puts()

**putchar()** แสดงตัวอักขระตัวเดียวออกทางจอภาพ

**puts()** แสดงข้อความสตริงออกทางจอภาพ

```
char d;  
putchar(d);
```

```
char d[20];  
puts(d);
```

# ฝึกการเขียนโปรแกรม

1. โปรแกรมคำนวณหาพื้นที่สามเหลี่ยมแสดงผลทางจอภาพ
2. โปรแกรมคำนวณค่าแรงรายวัน โดยรับค่าจำนวนชั่วโมงที่ทำงาน แสดงผลเงินที่ได้รับ โดยกำหนดให้ค่าแรงชั่วโมงละ 25 บาท
3. โปรแกรมคำนวณเกรดเฉลี่ยของภาคเรียนที่ 2 ในรายวิชาภาษาไทย คณิตศาสตร์ และวิทยาศาสตร์ โดยรับค่าเกรดทั้ง 3 รายวิชา และแสดงผลลัพธ์เป็นเกรดเฉลี่ย โดยกำหนดให้ (วิชาภาษาไทย 1.0 หน่วยกิต, วิชาคณิตศาสตร์ 2.0 หน่วยกิต, วิชาวิทยาศาสตร์ 1.5 หน่วยกิต)
4. โปรแกรมหาค่าเฉลี่ยของสามตัวเลขและแสดงผลลัพธ์
5. โปรแกรมหาค่าพื้นที่ของสี่เหลี่ยมลูกบาศก์ (Cube) และนำออกแสดงผล
6. โปรแกรมหาจำนวนดอกเบี้ยว I ที่ได้รับค่าเป็นจำนวนเงินฝาก P อัตราดอกเบี้ย R จำนวนวัน D และแสดงผล  
(ใช้สูตร  $I = P \cdot R \cdot D / 365$ )
7. โปรแกรมหาค่าตัวเลขที่ป้อนเข้ามาเป็นเลขคี่หรือเลขคู่แสดงผลให้ทราบ
8. โปรแกรมหาค่าที่มากที่สุดของสามตัวเลขที่ป้อนเข้ามาและนำออกแสดงผล