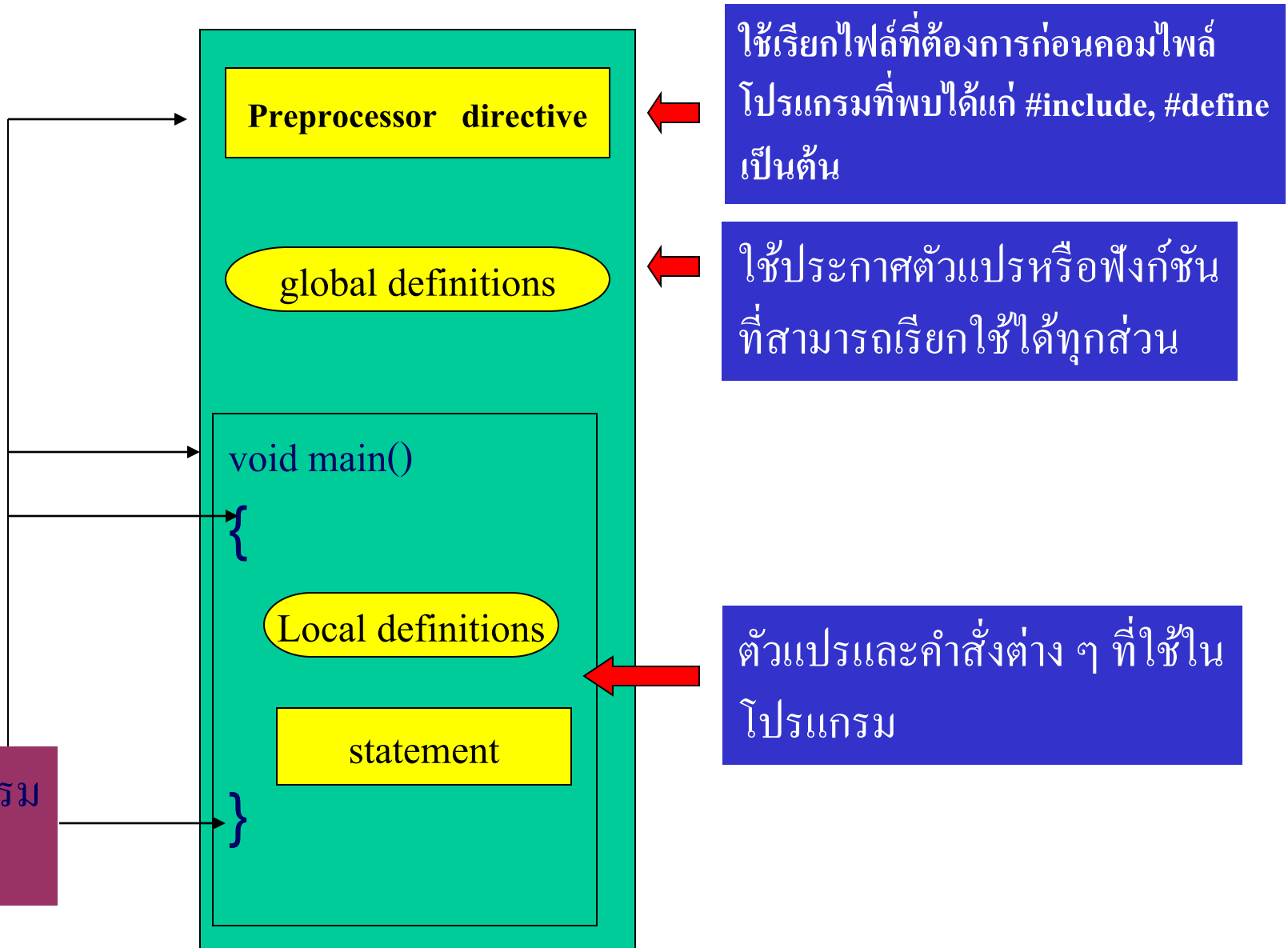


พื้นฐานภาษาซี

1. เริ่มต้นเขียนภาษาซี
2. ประเภทของข้อมูล
3. ตัวดำเนินการ
4. ฟังก์ชันรับและแสดงผลทางจอภาพ

โครงสร้างโปรแกรม



โครงสร้างโปรแกรม

```
1  #include<file.h>
   type function_name(type);
2  type variable;
   int main()
   {
       type variable;
       statement-1;
       ...
       statement-n;
       return 0;
   }
   type function_name(type variable)
   {
       statement-1;
       ...
       statement-n;
       return(var);
   }
```

Preprocessor Directive
Prototype Functions
Global Declarations

Header

Main Function

User-Defined Functions

Body

โปรแกรมแรก

```
#include <stdio.h>

void main()
{
    printf("Easy C \n");
    printf("Computer\n");
}
```

คำสั่ง include ให้นำไฟล์ stdio.h มาร่วมด้วย เพื่อเรียกใช้ฟังก์ชันต่าง ๆ หรืออาจเรียกไฟล์ .h อื่น ๆ อีกก็ได้ โดยไฟล์ที่เรียกจะต้องอยู่ใน “ “ หรือ < >

ฟังก์ชันในภาษาซี จบด้วยเครื่องหมาย ; ในที่นี่จะให้แสดงข้อความทางจอภาพ และขึ้นบรรทัดใหม่

ผลการรันโปรแกรม

Easy C

Computer

void ที่อยู่หน้า main จะบอกว่าเมื่อทำตั้งแต่ { ถึง } ไม่ต้องคืนค่า หรือไม่ต้องทำอะไรต่อไปอีก

คำอธิบายของโปรแกรม (Program Comments)

- ใช้เขียนส่วนอธิบายโปรแกรม (คอมเมนต์)
- ช่วยให้ผู้ศึกษาโปรแกรมภายหลังเข้าใจการทำงานของโปรแกรม
- ส่วนของคำอธิบายจะถูกข้ามเมื่อคอมไพล์โปรแกรม

การเขียนส่วนอธิบายโปรแกรม (comments)ทำได้ 2 วิธีคือ

// สำหรับคำอธิบายไปจนถึงท้ายบรรทัด

และ

/* คำอธิบาย */ ลักษณะการใช้เหมือนวงเล็บนั่นเอง

ไฟล์ .h ที่ตามหลัง include

ไฟล์นามสกุล .h เป็น Header file หรือไฟล์ส่วนหัวที่รวบรวมคำสั่งต่าง ๆ ของภาษาซีเอาไว้ เพื่อให้ผู้เขียนสามารถใช้ฟังก์ชันได้ มีอยู่หลายไฟล์เช่น

- **stdio.h** เก็บฟังก์ชันที่ใช้งานทั่วไป เช่น printf , scanf โปรแกรมส่วนมากจะใช้ไฟล์นี้
- **conio.h** เก็บฟังก์ชันควบคุมการแสดงผลต่าง ๆ
- **string.h** เก็บฟังก์ชันที่ใช้ในการประมวลผลกับข้อความ
- **math.h** เก็บฟังก์ชันที่ใช้ในการคำนวณทางคณิตศาสตร์ เช่น sin,cos,log

นักเรียนควรไปหาเพิ่มเติมว่ามีไฟล์ใดอีกให้เราเรียกมาใช้งาน.....

ตัวอย่างโปรแกรม

```
#include "stdio.h"
```

```
int feet, inches;
```

```
void main()
```

```
{
```

```
    feet = 6;
```

```
    inches = feet * 12;
```

```
    printf("Height in inches is %d",inches);
```

```
}
```

ประกาศตัวแปรสองตัวสำหรับ
เก็บเลขจำนวนเต็ม

กำหนดค่า 6 ให้กับตัวแปร feet ด้วย ;

พิมพ์ค่าออกทางจอเป็นเลขฐานสิบ

นำค่าในตัวแปร feet มาคูณกับ 12 แล้วเก็บในตัวแปร inches

ตัวแปร (variables)

ตัวแปร เป็นชื่อของหน่วยความจำที่ตำแหน่งต่าง ๆ ที่ผู้เขียน โปรแกรม กำหนด มีไว้สำหรับเก็บข้อมูลต่าง ๆ ระหว่างการทำโปรแกรม แบ่งตามประเภทได้ดังนี้

- ตัวแปรเก็บข้อมูลชนิดจำนวนเต็ม
- ตัวแปรเก็บข้อมูลชนิดจำนวนจริง (มีทั้งแบบ 32 บิต และ 64 บิต)
- ตัวแปรเก็บข้อมูลชนิดตัวอักษร
- ตัวแปรเก็บข้อมูลชนิดสายอักขระ (สตริง)

การประกาศตัวแปร

Data type variables_name

↑
ประเภทของข้อมูล

↑
ชื่อตัวแปร

Ex int x;

ประกาศตัวแปรชื่อ x เก็บจำนวนเต็ม

กฎการตั้งชื่อตัวแปร

- ประกอบด้วยตัว a ถึง z , 0 ถึง 9 และ _ เท่านั้น
- ตัวอักขระตัวแรกต้องเป็น a ถึง z และ _
- ห้ามใช้ชื่อเฉพาะ แล้วห้ามใช้คำสงวน
- ตัวพิมพ์ใหญ่ให้ตัวพิมพ์เล็ก ระบบจะมองว่ามีความหมายต่างกัน
- ความยาวตัวแปรไม่เกิน 31 ตัวอักขระ
- ควรตั้งชื่อให้สัมพันธ์สอดคล้องกับลักษณะงานที่จัดเก็บ

ชนิดข้อมูล (data type)

ชนิดข้อมูล	การประกาศ	ขนาด (Byte)	ช่วงของข้อมูล
ตัวอักษร	char	1	-128 ถึง 127
เลขจำนวนเต็ม	int	2	-32,768 ถึง 32,767
long integer	long	4	สองพันล้าน
จำนวนเต็ม ไม่คิดเครื่องหมาย	unsigned int	2	0 ถึง 65,535
เลขทศนิยม	float	4	1.2×10^{-38} - 3.4×10^{38}

ดูเพิ่มเติมในหนังสือ ถ้าหากคอมพิวเตอร์ต่างกันขนาดก็อาจต่างกันไปด้วย

คำสงวน (Reserved Word)

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

แบบฝึกหัด

จงตอบคำถามต่อไปนี้

1. จงตั้งชื่อตัวแปรและชนิดตัวแปรให้สอดคล้องกับข้อมูลที่กำหนดให้

ข้อมูล	ชื่อตัวแปร	ชนิดตัวแปร
รหัสนักเรียน		
ชื่อนักเรียน		
เพศ		
อายุ		
วันเกิด		
ที่อยู่		
รหัสไปรษณีย์		
เบอร์โทรศัพท์		

แบบฝึกหัด

2. จงตรวจสอบชื่อตัวแปรต่อไปนี้

ชื่อตัวแปร	ผิดหรือถูก	ผิดเพราะอะไร	ควรแก้ไขใหม่เป็น
1A	ผิด	ขึ้นต้นด้วยตัวเลข	A1
Hi			
Student 1			
Salary of week			
TelNo			
Age			
#F			
B!!!			
No.1			
x-ray			
-hightext-			

ตัวอย่างการตั้งชื่อตัวแปร

computer_room

n-sync

108dog

Year#

_good

ตัวอย่างการตั้งชื่อตัวแปร

computer_room	←	ถูกต้อง
n-sync	←	ผิดหลักการ เนื่องจากมีเครื่องหมาย – ในชื่อ
108dog	←	ผิดหลักการ เนื่องจากใช้ตัวเลขขึ้นต้น
Year#	←	ผิดหลักการ เนื่องจากมีเครื่องหมาย # อยู่ในชื่อ
_good	←	ถูกต้อง

การประกาศตัวแปร

จบด้วยเครื่องหมายอัฒภาค (;)

ชนิดข้อมูล ชื่อตัวแปร;

ถ้าหากมีตัวแปรหลายตัวให้ใช้เครื่องหมายจุลภาคคั่น

ชนิดข้อมูล ชื่อตัวแปร, ชื่อตัวแปร,.... ;

ตัวอย่าง

int count;	ประกาศตัวแปรชื่อ count สำหรับเก็บเลขจำนวนเต็ม
int m, n;	ประกาศตัวแปรชื่อ m และ n สำหรับเก็บเลขจำนวนเต็ม
int no = 10	ประกาศตัวแปรชื่อ no สำหรับเก็บจำนวนเต็มและให้เท่ากับ 10
float percent, total;	ประกาศตัวแปรชื่อ percent และ total สำหรับเก็บเลขทศนิยม

การกำหนดค่าให้กับตัวแปรตัวเลข

- การให้ค่ากับข้อมูลการกำหนดค่าพร้อมกับประกาศตัวแปร หรือประกาศแล้วจึงกำหนดก็ได้ ตัวอย่างเช่น
 - int no = 10; /* ประกาศตัวแปร no เป็นเลขจำนวนเต็มและให้เท่ากับ 10*/
 - int no; /* ประกาศตัวแปร no เป็นจำนวนเต็ม */
no = 10; /* ให้ no เท่ากับ 10 */
 - int num /* ประกาศตัวแปร num เป็นจำนวนเต็ม */
num = 0x10; /* ให้ num เท่ากับ 10 ฐานสิบหก */

เลขฐานสิบหกให้เริ่มต้นด้วย 0x

ตัวแปรชนิดอักขระ

```
char a, b;
```

```
a = 'T';
```

```
b = 'A';
```

← ประกาศตัวแปร a และ b เป็นตัวอักขระ

← กำหนดค่าตัวอักขระให้กับตัวแปรจะต้องอยู่ในเครื่องหมาย ‘ ‘

ตัวอย่าง

```
int a, b;      /* ประกาศตัวแปร a และ b สำหรับเก็บเลขจำนวนเต็ม */
```

```
char c, d;     /* ประกาศตัวแปร c และ d เก็บตัวอักขระ */
```

```
a = 3;
```

```
b = a;         /* เอาค่าในตัวแปร a ใส่ในตัวแปร b จะทำให้ b เท่ากับ 3 */
```

```
c = 'a';       /* ให้ค่าในตัวแปร c มีค่าเป็นตัวอักขระตัว a */
```

```
b = a + 4;     /* ให้ค่าในตัวแปร b เท่ากับ 7 */
```

```
b = 'a' + a;   /* ค่าในตัวแปร b จะเท่ากับรหัส ASCII ของ a บวกกับ 3 */
```

ตัวแปรชนิดข้อความ

ข้อความก็คือตัวอักษรหลาย ๆ ตัวมาต่อเรียงกันเรียกว่าสตริง (string) เมื่อมีการประกาศตัวแปรประเภทนี้ คอมพิวเตอร์จะนำตัวอักษรแต่ละตัวมาเก็บในหน่วยความจำแบบ char เรียงกันไปทีละตัว

รูปแบบ

ถ้าไม่ระบุก็ได้

```
char name[n] = "str"
```

ต่างจากภาษายุคใหม่ที่ใช้แบบอ้างอิง

name : ชื่อตัวแปร

n : ขนาดของข้อความ หรือจำนวนอักขระในข้อความ

str : ข้อความเริ่มต้นที่กำหนดให้กับตัวแปร โดยเขียนไว้ในเครื่องหมาย “ “

ตัวแปรชนิดข้อความ

ตัวอย่าง

ตัวแปรชื่อ name มีความยาวไม่เกิน 15 ตัวอักษร

char name[15] = "COMPUTER";

/* เก็บข้อความลงตัวแปร name */

char phone[15] = "0-2737-3000";

char age[3] = "20";

ตัวแปรชื่อ age มีความยาวไม่เกิน 3 ตัวอักษร

การประกาศค่าคงที่

ค่าคงที่เป็นค่าที่คงเดิมตลอดทั้งโปรแกรม แต่ตัวแปรจะมีค่าเปลี่ยนแปลงได้ระหว่างทำโปรแกรม

การประกาศค่าคงที่

1. ใช้คำสั่ง `const` ตามรูปแบบดังนี้

`const` ชนิดข้อมูล ชื่อตัวแปร = ค่าที่เก็บในตัวแปร;

Ex: `const int count = 120 ;`

2. ใช้ตัวประมวลผลก่อน ตามรูปแบบดังนี้

`#define` ชื่อค่าคงที่ ค่าคงที่ที่ต้องการเก็บ

Ex: `#define count 120`

`/* กำหนดให้ count เป็นค่าคงที่แบบ int มีค่า`

`เป็น 120*/`

แบบฝึกหัด

- ตัวแปรชื่อ age เป็นชนิดเก็บตัวเลขจำนวนเต็มอย่างเดียว
- ตัวแปร i,j เป็นตัวแปรเก็บข้อมูลที่เป็นตัวเลขจำนวนเต็ม
- ตัวแปร log, loop เก็บค่าที่เป็นตัวเลขทศนิยมอย่างยาว
- ค่าคงที่ m มีค่าเท่ากับ 90
- ค่าคงที่ n มีค่าเท่ากับ 2.5

นิพจน์ (Expression)

หมายถึงสิ่งที่ประมวลผลแล้วให้ค่าผลลัพธ์ออกมา มีความยากง่ายแตกต่างกันออกไป

นิพจน์ที่เป็นสัญลักษณ์

```
#define VAT 7  
#define PI 3.14159  
const int a = 35;
```

นิพจน์ตัวแปร

```
int count;  
float amount;  
char ch;
```

นิพจน์ที่เชื่อมโยงด้วยตัวดำเนินการ

```
3+5;  
score1 * 2 + score2 * 5 + score3 * 7;
```

ข้อความกำหนดค่า (Assignment statement)

เป็นการกำหนดค่าให้กับตัวแปร มีรูปแบบดังนี้

ตัวแปร = นิพจน์; ← จบด้วยเครื่องหมาย ;
↑
ใช้เครื่องหมายเท่ากับ

ตัวอย่าง

```
con = 10.5;
```

```
result = 25 * 6;
```

```
point = score1 * 2 + score2 * 5 + score3 * 8;
```

```
a = b = 0;      /*ให้ตัวแปร a เท่ากับตัวแปร b โดยมีค่าเป็นศูนย์ */
```


ตัวอย่าง

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

```
void main()
```

```
{
```

```
    int sum, total;
```

```
    sum = total = 0;
```

```
    printf(" sum = %d \n",sum);
```

```
    printf(" total = %d \n",total);
```

```
}
```

คำสั่งนี้จะได้เรียนต่อไป

```
sum = 0
```

```
total = 0
```

นิพจน์กำหนดค่า

สัญลักษณ์	ความหมาย
$>$	มากกว่า
$<$	น้อยกว่า
$>=$	มากกว่าหรือเท่ากับ
$<=$	น้อยกว่าหรือเท่ากับ
$==$	เท่ากับ
$!=$	ไม่เท่ากับ

นิพจน์กำหนดค่า

ความแตกต่างของเครื่องหมาย = และ ==

- เครื่องหมาย = เป็นตัวกำหนดค่า
- เครื่องหมาย == เป็นเครื่องหมายเปรียบเทียบ
- ตัวอย่างเช่น

K = 20;

หมายถึง เป็นการกำหนดค่าให้กับตัวแปร **K** ให้มีค่าเท่ากับ **20**

K == 20;

หมายถึง เป็นการตรวจสอบว่าค่า **K** มีค่าเท่ากับ **20** หรือไม่

เครื่องหมายและนิพจน์แบบตรรกศาสตร์

- เครื่องหมายและนิพจน์เปรียบเทียบแบบ

ตรรกศาสตร์

&&

หมายถึง

และ (and)

||

หมายถึง

หรือ (or)

!

หมายถึง

ไม่ (not)

ตัวอย่างเช่น

$x \ \&\& \ y$

จะได้ค่าความจริงเป็นจริงก็ต่อเมื่อ x และ y เป็นจริงทั้งคู่

$x \ || \ y$

จะได้ค่าความจริงเป็นเท็จก็ต่อเมื่อ x และ y เป็นเท็จทั้งคู่

$x \ ! \ y$

จะได้ค่าความจริงเป็นจริงก็ต่อเมื่อ x และ y ไม่เท่ากันทั้งคู่

ตัวดำเนินการทางคณิตศาสตร์

การคำนวณ	ตัวดำเนินการ	ตัวอย่าง	การทำงาน
บวก	+	$c = a + b;$	นำ a บวก b ผลลัพธ์เก็บใน c
ลบ	-	$c = a - b;$	นำ a ลบ b แล้วเก็บใน c
คูณ	*	$c = a * b;$	นำ a คูณ b แล้วเก็บใน c
หาร	/	$c = a / b;$	การหารต้องระวังจะขึ้น กับประเภทของข้อมูลด้วย
มอดุลัส	%	$c = a \% b;$	เป็นการหารที่เก็บเศษไว้ใน c

ตัวอย่าง

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์
+	การบวก	$6 + 8$	14
-	การลบ	$7 - 5$	2
*	การคูณ	$3 * 4$	12
/	การหาร	$8/2$	4
-	ลบ (ยูนิารีเครื่องหมายลบ)	-5	-5
%	โมดูลัส	$7 \% 2$	1
	(หาเศษเหลือจากการหาร)	$4 \% 2$	0

ตัวดำเนินการเอกภาค (unary operator)

เป็นการใช้ตัวดำเนินการกับตัวแปรตัวเดียว มีการใช้สองแบบคือ

1. ตัวดำเนินการเอกภาคเติมหลัง (postfix mode)

EX: `a++`; เพิ่มค่าครั้งละหนึ่งค่า

2. ตัวดำเนินการเอกภาคเติมหน้า (prefix mode)

EX: `++a`; เพิ่มค่าครั้งละหนึ่งค่า

ตัวอย่างการใช้งาน

การคำนวณ	ตัวดำเนินการ	ตัวอย่าง	การทำงาน
เพิ่มค่าที่ละหนึ่ง	++	$x++$	$x = x + 1$
		$++x$	$x = x + 1$
ลดค่าที่ละหนึ่ง	--	$x--$	$x = x - 1$
		$--x$	$x = x - 1$

$y = ++x;$



เพิ่มค่า x ขึ้นหนึ่งแล้วใส่ให้ตัวแปร y

$y = x++;$



นำค่า x ให้ตัวแปร y ก่อน จากนั้นเพิ่มค่า x ขึ้นหนึ่ง

ตัวดำเนินการประกอบ

เป็นการใช้ตัวดำเนินการหนึ่งตัวร่วมกับเครื่องหมายเท่ากับ ซึ่งจะเป็นตัวดำเนินการแบบลดรูป

ตัวดำเนินการประกอบ

ตัวอย่าง

การทำงาน

บวกแบบลดรูป



$+=$

$x += 5$

$x = x + 5$

$-=$

$x -= 5$

$x = x - 5$

$*=$

$x *= y$

$x = x * y$

$/=$

$x /= y$

$x = x / y$

$\%=$

$x \% = 5$

$x = x \% 5$

$+=$

$x += y/8$

$x = x + y / 8$

ตัวอย่างการใช้งาน

คำสั่ง	x	y
int x = 10, y = 20;	10	20
++x;	11	20
y = - -x;	10	10
x = x-- +y;	19	10
y = x- ++x;	20	0

```
#include <stdio.h>

void main( )
{
    int y=10, x=19;

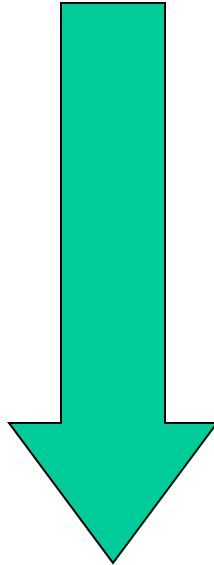
    y = x-++x;

    printf("y =%d\n",y);
    printf("x =%d\n",x);
}
```

ลำดับในการดำเนินการ

ถ้าหากมีการใช้ตัวดำเนินการประกอบและตัวดำเนินการเอกภาคหลายตัว จะมีลำดับทำงานดังนี้

1. ()
2. ++ --
3. * / %
4. + -
5. += *= /= -= %=



การเขียนนิพจน์ทางคณิตศาสตร์ในภาษาซี

นำข้อมูลหรือตัวแปรมาดำเนินการทางคณิตศาสตร์ ตรรกศาสตร์ ดังตัวอย่างต่อไปนี้

นิพจน์ทางคณิตศาสตร์ปกติ

$$x^2 + 4x + 2$$

$$\frac{x^2}{xy + 2}$$

$$\frac{a + b}{c - d}$$

นิพจน์ทางภาษาซี

$$x * x + 4 * x + 2$$

$$x * x / (x * y + 2)$$

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

การแปลงชนิดข้อมูล (type cast)

เป็นการเปลี่ยนแปลงข้อมูลชนิดหนึ่งให้เป็นอีกชนิดหนึ่ง

ตัวแปร = (ชนิดของข้อมูล) นิพจน์;

ข้อมูลใหม่ที่ต้องการให้กับตัวแปร

ตัวอย่าง

```
#include "stdio.h"
```

```
main()
```

```
{
```

```
    int x;
```

```
    x = 5.6 + 3.5;
```

/* จะได้ x = 9 */

```
    x = (int) 5.6 + (int) 3.5;
```

/* แปลง 5.6 ให้เป็น 5 แปลง 3.5 ให้เป็น 3 */

```
}
```

จะได้ x = 8

ฟังก์ชันรับข้อมูลและแสดงผล



ฟังก์ชันรับข้อมูล (Input Function)

- `scanf ()` รับข้อมูลจากคีย์บอร์ดมาเก็บในตัวแปรที่กำหนด
- `getchar()` รับข้อมูลอักขระตัวเดียว และต้องกด Enter ทุกครั้ง
- `getch()` รับข้อมูลอักขระตัวเดียว ไม่ต้องกด Enter และไม่แสดงผล
- `getche()` รับข้อมูลอักขระตัวเดียวเหมือน `getch()` แต่จะแสดงผลด้วย
- `gets()` รับข้อความ (string) มาเก็บในตัวแปรที่กำหนด

scanf()

scanf(control string ; argument list)

- control string เป็นรหัสรูปแบบข้อมูล (format code)
- argument list ตัวแปรที่ใช้เก็บข้อมูล ใช้เครื่องหมาย & นำหน้าชื่อ แต่
ถ้ารับข้อมูลเป็นสตริงไม่ต้องมีเครื่องหมายนี้

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

getchar()

getchar ();

หรือ char_var = getchar();

char_var เป็นตัวแปรชนิด char ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่าน
คีย์บอร์ด

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

```
void main( )
```

```
{
```

```
    char c;
```

```
    //clrscr();
```

```
    printf("Enter a single character :");
```

```
    c = getchar( );
```

```
    printf("You type a character is .... %c\n",c);
```

```
}
```

getch()

getch ();

หรือ char_var = getch();

char_var เป็นตัวแปรชนิด char ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่านคีย์บอร์ด

ฟังก์ชันนี้ใช้รับอักขระตัวเดียวจากคีย์บอร์ด ไม่ต้องกด Enter ข้อมูลที่รับเข้าไปจะไม่แสดงผล

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

```
void main( )
```

```
{
```

```
    char d;
```

```
    clrscr();
```

```
    printf("Enter a single character :");
```

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

```
    printf("You type a character is .... %c\n",d);
```

```
}
```

getche()

getche ();

หรือ char_var = getche();

char_var เป็นตัวแปรชนิด char ที่ใช้เก็บข้อมูลตัวอักขระที่ป้อนผ่านคีย์บอร์ด

ฟังก์ชันนี้ใช้รับอักขระตัวเดียวจากคีย์บอร์ด ไม่ต้องกด Enter ข้อมูลที่รับเข้าไปจะแสดงผลออกมาด้วย

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

```
void main( )
```

```
{
```

```
    char e;
```

```
    clrscr();
```

```
    printf("Enter a single character :");
```

```
    e = getche( );
```

```
    printf("You type a character is .... %c\n",e);
```

```
}
```


gets()

```
gets(string_var);
```

string_var เป็นตัวแปรสตริงที่ใช้เก็บข้อความ (string constant)

ฟังก์ชันนี้ใช้รับข้อมูลจากคีย์บอร์ดมาเก็บในตัวแปรสตริง

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

```
void main( )
```

```
{
```

```
    char name[30];
```

```
    int age
```

```
    printf("Enter your name : ");
```

```
    gets(name);
```

```
    printf("Enter your age : ");
```

```
    scanf("%d",&age);
```

```
    printf("Your name is ...%s\n",name)
```

```
    printf("Your are %d years old. \n",age);
```

```
}
```

สรุปการใช้ **input function**

- รับข้อมูล string ควรใช้ฟังก์ชัน `gets()` หรือ `scanf()`
- รับข้อมูลตัวเลขควรใช้ฟังก์ชัน `scanf()` ถ้าหากป้อนข้อมูลแล้วเว้นวรรค โปรแกรมจะมองว่าเป็นตัวแปรมากกว่าหนึ่งตัว
- รับตัวเลขหนึ่งตัวหรือตัวอักขระหนึ่งตัว ไม่แสดงทางจอภาพ ใช้ `getch()` แต่ถ้าต้องการให้แสดงทางจอภาพใช้ `getche()`
- การใช้ `scanf()` รับข้อมูลติดกันมากกว่า 2 ครั้งอาจเกิดข้อผิดพลาดได้

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 ถ้าหากมีมากกว่า
หนึ่งตัวให้ใช้ comma คั่นระหว่างแต่ละตัว

ตัวอย่าง

```
int x1 = 43, x2 = 0x77 , x3 = 0573;
```

```
float y1 = -764.512 , y2 = 1.25e02;
```

```
char z = 'A';
```

```
char name[10] = "COMPUTER ";
```

```
main()
```

```
{
```

```
    printf("%d ",x1);
```

```
    printf("%x %o ",x2,x3);
```

```
    printf("%f %e ",y1,y2);
```

```
    printf("%c %s ",z,name);
```

```
}
```

การกำหนดรูปแบบทศนิยม

รูปแบบ

%.nf



จำนวนจุดทศนิยมที่ต้องการแสดงผล

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

```
float x1 = 17.765 ,x2 = 1.578e+03;
```

```
main()
```

```
{
```

```
    printf("%.4f ",x1);
```

```
    printf("%.1f ",x1);
```

```
    printf("%.2e ",x2);
```

```
    printf("%.1e ",x2);
```

```
}
```

การกำหนดพื้นที่แสดงผล

รูปแบบ %m.nf

m : จำนวนพื้นที่แสดงผล

n : จำนวนอักขระที่ต้องการนำมาแสดงผล

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

```
char message[ ] = "C Language";
```

```
main()
```

```
{
```

```
    printf("%12.6s",message);
```

```
}
```


รหัสควบคุมการแสดงผล

- \n ขึ้นบรรทัดใหม่
- \t เว้นช่องว่างเป็นระยะ 1 tab หรือหกตัวอักขระ
- \r ให้เคอร์เซอร์ขึ้นบรรทัดใหม่
- \f เว้นช่องว่างเป็นระยะหนึ่งหน้าจอ

ตัวอย่างการแสดงผล

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

```
#define PUBLISH "Infopress"
```

```
int x1 = 14, x2 = 5;main()
```

```
{
```

```
printf("This is the book from %s\n",PUBLISH);
```

```
printf("Sum of %d + %d = \t%d\n",x1,x2,x1+x2);
```

```
}
```

putchar() แสดงอักขระตัวเดียว

```
putchar(char);
```

↑ เป็นตัวแปรชนิด char หรือเขียนภายในเครื่องหมาย ‘ ‘

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

```
char first = 'A';
```

```
main()
```

```
{
```

```
    putchar(first);
```

```
    printf("\n");
```

```
    putchar('T');
```

```
}
```