

บทที่ 3 ชนิดของข้อมูลในภาษาซี

คำสั่งรับข้อมูล และการคำนวณ

ศท211 วิทยาการคอมพิวเตอร์ 1

โดย อาจารย์ภาณุวัฒน์ เมฆะ

สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

วัตถุประสงค์

- รู้จักและเข้าใจข้อมูลประเภทต่างๆ
- สามารถสร้างตัวแปรและใช้งานตัวแปรได้อย่างถูกต้อง
- สามารถแสดงข้อความออกทางจอภาพและรับข้อมูลจากคีย์บอร์ดได้
- สามารถใช้ Operator ทางคณิตศาสตร์ได้
- เขียนโปรแกรมประมวลผลตัวอักษรได้

3.0 ตัวแปรในภาษาซี

- การประกาศตัวแปรมี 2 ลักษณะ ได้แก่ ตัวแปร โทบอล และ ตัวแปร โลคอล ซึ่งมีรูปแบบการประกาศตัวแปรที่เหมือนกัน แต่จะมีคุณสมบัติต่างกัน โดยจะเห็นได้ชัดเจนเมื่อมีการสร้างฟังก์ชันมาใช้งาน
- เนื้อหาส่วนต้น จะใช้งานเฉพาะตัวแปรชนิดโลคอล โดยจะประกาศตัวแปรในส่วนเริ่มต้นของฟังก์ชัน

3.1 ชนิดของข้อมูลในภาษาซี

- เป็นพารามิเตอร์ที่ใช้งานในภาษาซี เพื่อกำหนดลักษณะฟังก์ชัน ต้นแบบ และ ตัวแปรที่ใช้งานใน โปรแกรม มี 4 ชนิด

void	ข้อมูลชนิดว่างเปล่า
int	ข้อมูลชนิดจำนวนเต็ม
float	ข้อมูลชนิดจำนวนทศนิยม
char	ข้อมูลชนิดอักขระ

3.1.1 ข้อมูลชนิดว่างเปล่า / **void**

void เป็นพารามิเตอร์ที่ใช้งานในส่วนของ ฟังก์ชัน โพรโททรไทป์ การสร้าง และใช้งานฟังก์ชัน เพื่อแสดงให้รู้ว่าฟังก์ชันที่สร้างขึ้นมาไม่มีการส่ง หรือรับค่าจากการเรียกใช้งานฟังก์ชัน

3.1.2 ข้อมูลชนิดจำนวนเต็ม / integer

int เป็นพารามิเตอร์หลักที่ใช้กับข้อมูลชนิดจำนวนเต็ม โดยมีการใช้งาน 5 รูปแบบดังนี้

unsigned int ข้อมูลชนิดจำนวนเต็มไม่กิดเครื่องหมายขนาด 2 bytes

short int ข้อมูลชนิดจำนวนเต็มขนาด 2 bytes

int ข้อมูลชนิดจำนวนเต็มขนาด 2 bytes หรือ 4 byte

(ในคอมพิวเตอร์ 32 bit ตัวแปร int มีขนาด 4 byte แต่ในคอมพิวเตอร์ 16 bit ตัวแปร int มีขนาด 2 byte)

unsigned long ข้อมูลชนิดจำนวนเต็มไม่กิดเครื่องหมายขนาด 4 bytes

long ข้อมูลชนิดจำนวนเต็มขนาด 4 bytes

3.1.3 ข้อมูลชนิดจำนวนทศนิยม / float

float เป็นพารามิเตอร์หลักที่ใช้กับข้อมูลชนิดจำนวนทศนิยม
โดยมีการใช้งาน 3 รูปแบบดังนี้

float	ข้อมูลชนิดจำนวนทศนิยมขนาด 4 byte
double	ข้อมูลชนิดจำนวนทศนิยมขนาด 8 byte
long double	ข้อมูลชนิดจำนวนทศนิยมขนาด 10 byte

3.1.4 ข้อมูลชนิดอักขระ / **character**

char เป็นพารามิเตอร์ที่ใช้งานเกี่ยวกับตัวอักษร และข้อความในภาษาซี โดยมีการกำหนดค่าอักขระ โดยให้อยู่ในเครื่องหมาย single quote (' . . . ') เช่น 'C' , 'o' , 'm' , '1'

อักขระพิเศษบางตัวไม่สามารถกำหนดค่าให้ได้โดยตรง แต่ใช้ค่ารหัส ASCII เช่นอักขระควบคุมการแสดงผลการขึ้นต้นบรรทัดใหม่ใช้ '**\n**' เป็นต้น โดยมีรูปแบบการใช้งาน 2 รูปแบบ

unsigned char ข้อมูลชนิดอักขระ ไม่คิดเครื่องหมาย

char ข้อมูลชนิดอักขระปกติ

3.1.5 ตารางแสดงขนาดและขอบเขตชนิดของข้อมูล

ชนิดของตัวแปร	ขนาด	ค่าต่ำสุด	ค่าสูงสุด
unsigned char	8 bit	0	255
char	8 bit	-128	127
unsigned int	16 bit	0	65,535
short int	16 bit	-32,768	32,767
int	16 bit	-32,768	32,767
unsigned long	32 bit	0	4,294,967,295
long	32 bit	-2,147,483,648	2,147,483,647
float	32 bit	3.4×10^{-38}	3.4×10^{38}
double	64 bit	1.7×10^{-308}	1.7×10^{308}
long double	80 bit	3.4×10^{-4932}	3.4×10^{4932}

3.2 รูปแบบการประกาศตัวแปรในภาษาซี

3.2.1 การประกาศตัวแปรแบบไม่กำหนดค่าเริ่มต้น

```
type    var1;
```

```
type    var1, var2, ..., varN;
```

type คือ ชนิดของข้อมูลที่จะกำหนดให้กับตัวแปร

varX คือ ชื่อของตัวแปรที่จะตั้ง

ตัวอย่างการประกาศตัวแปร

```
int    number;
```

```
int    a, b, c;
```

```
float  real;
```

```
float  point1, point2;
```

```
char   choice;
```

```
char   ch1, ch2;
```

3.2.2 รูปแบบการประกาศและกำหนดค่าตัวแปรในภาษาซี

```
type var1 = value1;
```

```
type var1 = value1, varN = valueN;
```

type คือ ชนิดของข้อมูลที่จะกำหนดให้กับตัวแปร

varX คือ ชื่อของตัวแปรที่จะตั้ง

valueX คือ ค่าของตัวแปรที่ต้องการกำหนดให้

ตัวอย่างการประกาศและกำหนดค่าตัวแปร

```
int    number = 25;
```

```
int    a = 1, b = 2, c = 3;
```

```
float  real = 99.99;
```

```
float  point1 = 45.2, point2 = 30;
```

```
char   choice = 'a';
```

```
char   ch1 = 'o', ch2 = 'z';
```

หลักการตั้งชื่อตัวแปรในภาษาซี

- เริ่มต้นด้วยอักษร **A-Z, a-z** หรือ เครื่องหมาย **_** เท่านั้น
- ภายในตัวแปร ห้ามมีช่องว่าง
- ภายในตัวแปรประกอบด้วยอักษร **A-Z, a-z**, ตัวเลข **0-9** หรือ เครื่องหมาย **_** เท่านั้น
- การใช้อักษรตัวใหญ่และตัวเล็ก มีความแตกต่างกัน
- ห้าม ใช้คำสงวนเป็นชื่อตัวแปร
- ควรตั้งชื่อตัวแปรให้สัมพันธ์กับค่าที่ใช้เก็บ

คำสงวน (Reserved word) ในภาษาซี

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			

Hint : ในโปรแกรม Turbo C++ Reserved Word จะเป็นตัวหนาสีดำ

ตัวอย่างการตั้งชื่อตัวแปร

//ตั้งชื่อตัวแปรถูกต้อง

```
int _money;
```

```
float salary_ot;
```

```
char M11223_223;
```

```
double GOTO_data;
```

//ตั้งชื่อตัวแปรผิด

```
int $money;
```

```
float static;
```

```
char 1M1223_223;
```

```
double GOTO-data;
```


3.3 ตัวแปรชนิดข้อความในภาษาซี

- ในภาษาซีจะ ไม่มีข้อมูลชนิดข้อความโดยเฉพาะ ซึ่งในการเขียนโปรแกรมส่วนใหญ่จำเป็นต้องมีการรับข้อมูลที่เป็นข้อความ เราสามารถใช้ตัวแปรชนิดอักขระหลายๆ ตัวมาใช้งานได้ในระดับหนึ่ง แต่ยังไม่สะดวกเมื่อข้อความมีความยาวมาก

ตัวอย่าง เช่น ต้องการใช้ข้อความว่า Hello สามารถใช้ตัวแปรชนิดอักขระ 5 ตัวแทน

```
char ch1='H' , ch2='e' , ch3='l' ,  
      ch4='l' , ch5='o' ;
```

ตัวแปรชนิดข้อความในภาษาซี

- ตัวแปรชนิดข้อความในภาษาซี คือ การนำอักขรมาเรียงต่อกัน ดังนั้นสามารถสร้างตัวแปรชนิดอักขระเรียงต่อกันหลาย ๆ ตัว ให้เป็นตัวแปรชนิดแถวลำดับ ทำให้สามารถใช้เก็บข้อมูลชนิดข้อความได้
- โดยตัวแปรชนิดข้อความในภาษาซีจะอยู่ในเครื่องหมาย Double quote " "

3.3.1 รูปแบบการประกาศตัวแปรชนิดข้อความ

```
char    var1 [M1] ;
```

```
char    var1 [M1] , var2 [M2] ;
```

varX คือ ชื่อตัวแปร

MX คือ จำนวนของอักขระที่จะใช้เก็บบวกด้วย 1

การใช้ตัวแปรแถวลำดับชนิดอักขระเป็นตัวแปรข้อความ ในภาษาซีกำหนดไว้ว่าตัวสุดท้ายของตัวแปรแถวลำดับคือ \0

\0 → Null character

3.3.2 รูปแบบการประกาศตัวแปรและกำหนดค่า

```
char    var [M] = "??...?";
```

```
char    var [M] = {'?', '?', ..., '?'};
```

```
char    var [] = "??...? ";
```

var คือ ชื่อตัวแปร

M คือ จำนวนของอักขระที่จะใช้เก็บบวกด้วย 1

? คือ อักขระที่จะกำหนดค่าให้ข้อความ จำนวน $m-1$ ตัว

ตัวอย่างการประกาศและกำหนดค่าตัวแปร

```
char subject[12] = "Programming";  
char nick[4] = "Com";  
char nick_1[4] = {'C','o','m','\0'};  
char name[] = "Somsak";
```

subject[12]

P r o g r a m m i n g \0

nick[4]

C o m \0

name []

S o m s a k \0

nick_1[4]

C o m \0

3.3.3 รูปแบบการอ้างอิงอักขรในตัวแปรข้อความ

```
variable[N]
```

variable คือชื่อตัวแปร

N คือลำดับอักขรที่จะอ้างอิงในตัวแปรข้อความ

เริ่มนับอักขรตัวแรกเป็นตำแหน่งที่ 0

ตัวอย่างการอ้างอิงอักขรในตัวแปรข้อความ

```
char    subject[12] = "Programming";
```

subject[12]

P	r	o	g	r	a	m	m	i	n	g	\0
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]

subject[0] → 'P'

subject[1] → 'r'

subject[2] → 'o'

subject[10] → 'g'

3.4 การใช้งานตัวแปรร่วมกับคำสั่ง `printf`

- ในคำสั่ง `printf` มีส่วนแสดงชนิดข้อมูล ซึ่งตัวแปรที่ใช้งานก็เป็นส่วนย่อยของข้อมูลชนิดต่าง ๆ

```
printf (format-string, data-list);
```

- จึงสามารถใช้ตัวแปรแทนในส่วนของ `data-list` ได้ เช่น

```
#include<stdio.h>
int main()
{
    printf ("GPA : %.2f",3.5);
    return 0;
}
```

```
#include<stdio.h>
int main()
{
    float g = 3.5;
    printf ("GPA : %.2f",g);
    return 0;
}
```



ตัวอย่างการใช้งานคำสั่ง `printf` โดยไม่ต้องใช้ตัวแปร

ต้องการเขียนโปรแกรมแสดงคำว่า **Programming** จำนวน 5 บรรทัดที่จอคอมพิวเตอร์

```
#include <stdio.h>
int main()
{
    printf ("%s\n", "Programming");
    printf ("%s\n", "Programming");
    printf ("%s\n", "Programming");
    printf ("%s\n", "Programming");
    printf ("%s\n", "Programming");
    return 0;
}
```

หากต้องการ
เปลี่ยนจากคำว่า
Programming

เป็น **Kmitl**

จะทำอย่างไร 

ตัวอย่างการใช้งานคำสั่ง `printf` โดยใช้ตัวแปร

ต้องการเขียนโปรแกรมแสดงคำว่า **Programming** จำนวน 5 บรรทัดที่จอคอมพิวเตอร์ โดยใช้ตัวแปรข้อความ

```
#include <stdio.h>
int main()
{
    char name[20]="Kmitl";
    printf ("%s\n",name);
    printf ("%s\n",name);
    printf ("%s\n",name);
    printf ("%s\n",name);
    printf ("%s\n",name);
    return 0;
}
```

หากต้องการ
แสดงข้อความ
ตามผู้ใช้กำหนด
จะอย่างไร?



3.5 การรับค่าด้วยคำสั่ง **scanf**

- โปรแกรมโดยทั่วไปต้องมีการรับค่าข้อมูลจากผู้ใช้โปรแกรม เพื่อนำมาหาผลลัพธ์ตามกระบวนการทำงานของ โปรแกรม หรือตามความต้องการของผู้ใช้งาน
- คำสั่งที่ใช้สำหรับการรับค่าในภาษาซีมีหลายคำสั่ง แต่ที่สามารถใช้งานได้ครอบคลุมและนิยมใช้กัน คือ คำสั่ง `scanf`

3.5.1 รูปแบบของคำสั่ง `scanf`

```
scanf (format-string, address-list) ;
```

`format-string` คล้ายในคำสั่ง `printf` แต่จะมีเฉพาะส่วนแสดงชนิดข้อมูล และอยู่ในเครื่องหมาย " "

`address-list` คือ ตำแหน่งของตัวแปรที่ต้องการเก็บข้อมูลไว้ (การใช้งานตำแหน่งของตัวแปรจะใช้เครื่องหมาย & นำหน้าชื่อตัวแปร ยกเว้นตัวแปรชนิดข้อความ)

หมายเหตุ คำสั่ง `scanf` ต้องเรียกใช้ Preprocessor Directive `#include<stdio.h>`

ตัวอย่างการใช้คำสั่ง `scanf`

```
#include<stdio.h>
int main()
{
    float point;
    char name[20];
    printf ("Enter your name : ");
    scanf ("%s",name);
    printf ("Enter your point : ");
    scanf ("%f",&point);
    return 0;
}
```

ตัวแปรข้อความ

ไม่ต้องมี & หน้าตัวแปร

ตัวแปรทศนิยม

ต้องมี & หน้าตัวแปร

ตัวอย่างการใช้คำสั่ง `printf` และ `scanf`

```
#include<stdio.h>
int main()
{
    char  first[20],last[20];

    printf ("Enter your name and surname : ");
    scanf ("%s %s",first,last);
    printf ("Hi %s %s\nHow are you?",first,last);
    return 0;
}
```

การรับตัวแปรมากกว่า 1 ตัวในคำสั่งเดียว
ต้องป้อนข้อมูลให้มีรูปแบบเหมือนกัน

```
Enter your name and surname : Somsak Jaidee
Hi Somsak Jaidee
How are you?
```

3.5.2 การใช้คำสั่ง **scanf** รับข้อความที่มีการเว้น (spacebar)

- คำสั่ง `scanf` ไม่สามารถใช้รับข้อความที่มีการเว้นเพื่อเก็บในตัวแปรข้อความตัวเดียวได้ (เมื่อใช้ `%s`)
- เราสามารถใช้คำสั่ง `scanf` เพื่อให้รับข้อความที่มีการเว้นไปเก็บในตัวแปรชนิดข้อความได้ดังนี้

```
scanf ("%[^\\n]", string);
```

ตัวอย่างการใช้คำสั่ง `printf` และ `scanf`

```
#include<stdio.h>
int main()
{
    char name[40];
    float gpa;
    printf ("Enter your name : ");
    scanf ("%[^\\n]", name);
    printf ("Enter your GPA : ");
    scanf ("%f", &gpa);
    printf ("Name : %s\\n", name);
    printf ("Gpa : %f", gpa);
    return 0;
}
```

(Inactive C:\TCWIN45\BIN\NONAME00.EXE)

```
Enter your name : Com eng
Enter your GPA : 3.7
Name : Com eng
Gpa : 3.700000
```



3.6 เครื่องหมายคำนวณทางคณิตศาสตร์

เครื่องหมาย	การทำงาน	ตัวอย่าง
+	บวก	<code>ans = a + b;</code>
-	ลบ	<code>ans = a - b;</code>
*	คูณ	<code>ans = a * b;</code>
/	หาร	<code>ans = a / b;</code>
%	โมดูลัส (modulo)	<code>ans = a % b;</code>

การกำหนดค่าให้ตัวแปร

- การกำหนดค่าให้กับตัวแปรในภาษาซีใช้เครื่องหมาย = โดยการ
ทำงานจะนำค่าที่อยู่ทางขวามือ (จำนวน อักขระ ข้อความ ค่าจาก
ตัวแปร หรือผลลัพธ์จากฟังก์ชัน) ให้กับตัวแปรที่อยู่ทางซ้ายมือ

```
variable = value;
```

```
num1 = 99;
```

```
n = num % 10;
```

```
point = mid + final;
```

```
ch = '9';
```

```
ans = pow(x, y);
```

```
str = "Com";
```

3.6.1 การเพิ่มลดค่าตัวแปร

เครื่องหมาย	การทำงาน	ตัวอย่าง	ขั้นตอนการทำงาน	
++	เพิ่มค่าทีละ 1 (Increment)	x++ ; ++x ;	เพิ่มค่า x ขึ้น 1	
		y = ++x ;	เพิ่มค่า x ขึ้น 1	กำหนดค่าให้ y
		y = x++ ;	กำหนดค่าให้ y	เพิ่มค่า x ขึ้น 1
--	ลดค่าทีละ 1 (Decrement)	x-- ; --x ;	ลดค่า x ลง 1	
		y = --x ;	ลดค่า x ลง 1	กำหนดค่าให้ y
		y = x-- ;	กำหนดค่าให้ y	ลดค่า x ลง 1

3.6.2 เครื่องหมายแบบลดรูป

เครื่องหมาย	ตัวอย่างการใช้งาน	ตัวอย่างรูปแบบเต็ม
<code>+=</code>	<code>y += x;</code>	<code>y = y + x;</code>
<code>-=</code>	<code>y -= x;</code>	<code>y = y - x;</code>
<code>*=</code>	<code>y *= x;</code>	<code>y = y * x;</code>
<code>/=</code>	<code>y /= x;</code>	<code>y = y / x;</code>
<code>%=</code>	<code>y %= x;</code>	<code>y = y % x;</code>

3.6.3 ตัวดำเนินการ & ตัวถูกดำเนินการ

Type 1	Operator	Type 2	Result	Exam	
int	+, -, *, /	int	int	3*3	9
				19/2	9
int	+, -, *, /	float	float	3*3.0	9.000000
				19/2.0	9.500000
float	+, -, *, /	int	float	3.0*3	9.000000
				19.0/2	9.500000
float	+, -, *, /	float	float	3.0*3.0	9.000000
				19.0/2.0	9.500000
int	%	int	int	7%4	3
				8%4	0

3.6.4 นิพจน์ในภาษาซี

ลำดับความสำคัญ	เครื่องหมาย
1	()
2	!, ++, --, (type)
3	*, /, %
4	+, -
5	<, <=, >, >=
6	==, !=
7	&&
8	
9	*=, /=, %=, +=, -=

ตัวอย่าง แสดงลำดับความสำคัญสัญลักษณ์ทางคณิตศาสตร์

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

```
int main()
```

```
{
```

```
    int a = 10;
```

```
    printf(" %d ", 10*2*3-10/7);
```

```
    printf(" %d ", 10*2*(3-10)/7);
```

```
    printf(" %d ", 10*2>2+3);
```

```
    printf(" %d ", 10*2<2+3);
```

```
    return 0;
```

```
}
```

```
//ผลลัพธ์
```

```
58 -20 1 0
```

3.7 คำสั่งแสดงผล และรับข้อมูลอื่นๆ

```
putchar (ch) ;
```

```
puts (str) ;
```

```
ch = getchar () ;
```

```
ch = getch () ;
```

```
gets (str) ;
```


รูปแบบและการใช้คำสั่ง `getchar`

- เป็นคำสั่งที่ใช้รับข้อมูลชนิดอักขระจากผู้ใช้งานเพียงตัวเดียว โดยเมื่อป้อนข้อมูลแล้วต้องกด Enter
- คำสั่ง `getchar` มีรูปแบบการใช้งานคำสั่งดังนี้

```
ch = getchar();
```

ch คือตัวแปรชนิดข้อความที่ต้องการเก็บข้อมูลไว้

หมายเหตุ คำสั่ง `getchar` ต้องเรียกใช้ Preprocessor Directive `#include<stdio.h>`

รูปแบบและการใช้คำสั่ง `getch`

- เป็นคำสั่งที่ใช้รับข้อมูลชนิดอักขระจากผู้ใช้งานเพียงตัวเดียว โดยเมื่อป้อนข้อมูลแล้ว โปรแกรมจะทำงานคำสั่งต่อไปทันที และจะ ไม่แสดงอักขระที่พิมพ์ไป
- คำสั่ง `getch` มีรูปแบบการใช้งานคำสั่งดังนี้

```
ch = getch ( ) ;
```

ch คือตัวแปรชนิดข้อความที่ต้องการเก็บข้อมูลไว้

หมายเหตุ คำสั่ง `getch` ต้องเรียกใช้ Preprocessor Directive `#include<conio.h>`

รูปแบบและการใช้คำสั่ง `gets`

- เป็นคำสั่งที่ใช้รับข้อมูลชนิดข้อความจากผู้ใช้งาน โดยสามารถใส่ข้อมูลที่มีการเว้นช่องว่างภายในได้
- คำสั่ง `gets` มีรูปแบบการใช้งานคำสั่งดังนี้

```
gets (str) ;
```

`str` คือตัวแปรชนิดข้อความที่ต้องการแสดงผล

หมายเหตุ คำสั่ง `gets` ต้องเรียกใช้ Preprocessor Directive `#include<stdio.h>`

ตัวอย่างคำสั่งแสดงผล และรับข้อมูล

```
char ch1, ch2;  
printf ("Enter Character 1 : ");  
ch1 = getchar();  
printf ("Enter Character 2 : ");  
ch2 = getch();  
puts ("\n**** Output ****");  
printf ("Char 1 = %c\nChar 2 = %c", ch1, ch2);
```

```
Enter Character 1 : J  
Enter Character 2 :  
**** Output ****  
Char 1 = J  
Char 2 = 0
```

พิมพ์ O แต่ไม่แสดงออกมา



3.8 คำถามท้ายบท

1. ชนิดข้อมูลแบบข้อความเหมือนหรือแตกต่างจากชนิดข้อมูลแบบอักขระอย่างไร
2. จากตัวอย่างโปรแกรมต่อไปนี้

```
#include <stdio.h>
int main() {
char name[20];
int age;
printf("Enter your name: "); scanf("%s",name);
printf("Enter your age "); scanf("%d",age);
printf("Your name is : %s\n Your age is : %d",name,age);
return 0; }
```

เมื่อ Run แล้วจะได้ผลอย่างไร

คำถามท้ายบท (ต่อ)

3. จากโปรแกรมต่อไปนี้

```
#include <stdio.h>
int main()
{
printf("%d", (40/4*3+4*5/2));
return 0;
}
```

เมื่อ Run แล้วจะได้ผลอย่างไร

ชีวะสะเด หมายถึง สวัสดิ์ดีในภาษากัมพูชา

