



Physical Computing

ครั้งที่ 10. Structure

Variable Types in C

- **Variable Types:**



8 bits

byte
char



16 bits

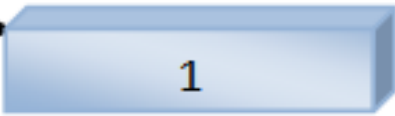
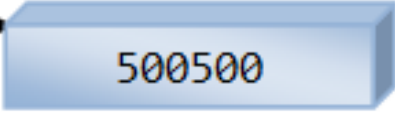
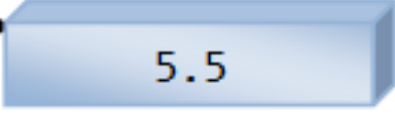
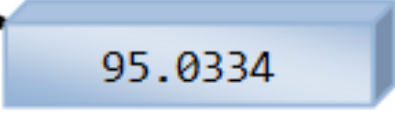
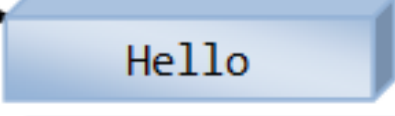
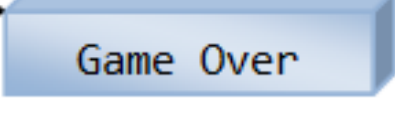
int
unsigned int



32 bits

long
unsigned long
float

Variables in C

TYPE	NAME	VALUE	
int	number	→ 	Stored only Integer
int	sum	→ 	Stored only Integer
double	radius	→ 	Stored only floating-point number
double	area	→ 	Stored only floating-point number
String	greeting	→ 	Stored only texts
String	statusMsg	→ 	Stored only texts

A *variable* has a **name**, stores a **value** of the declared **type**.

วัตถุประสงค์

- รู้จักและเข้าใจ ตัวแปรชนิดโครงสร้าง
- สามารถประกาศสร้าง ตัวแปรชนิดโครงสร้าง ได้
- สามารถประกาศสร้าง อาร์เรย์ของตัวแปรชนิดโครงสร้าง
- สามารถประกาศสร้าง ตัวแปรชนิดโครงสร้าง ช้อนตัวแปรชนิดโครงสร้าง
- สามารถใช้งานคำสั่งต่างๆ กับตัวแปรชนิดโครงสร้างได้

1.ความสำคัญของการใช้งาน ตัวแปรชนิดโครงสร้าง

- ในการเขียนโปรแกรมบางกรณี เมื่อนำข้อมูลตัวแปรบางตัวที่สามารถจัดรวมกันเป็นกลุ่ม (โครงสร้าง) มารวมกัน จะทำให้ใช้งานตัวแปรได้สะดวกถูกต้องยิ่งขึ้น เช่น ข้อมูลของบริษัท ข้อมูลของนักศึกษา ข้อมูลของบุคลากร ข้อมูลของตัวละคร

ตัวอย่างเช่น ข้อมูลของตัวละคร ประกอบด้วย

- ชื่อ (name) นามสกุล (surname) อายุ (age)
- วันเกิด/เดือนเกิด (b_date/b_month) เพศ (gender)
- ส่วนสูง (height) น้ำหนัก (weight) กรุ๊ปเลือด (blood)

HUNTER HUNTER



name : Kurapica
surname : Terminada
age : 17
b_date : 4
b_month : April
gander : male
height : 171
weight : 59
blood : AB



name : Gon
surname : Freecss
age : 12
b_date : 5
b_month : May
gander : male
height : 154
weight : 49
blood : O



name : Killua
surname : Zoaldyeck
age : 12
b_date : 15
b_month : June
gander : male
height : 158
weight : 49
blood : B

- ถ้าเขียนโปรแกรมโดยไม่ใช้ ตัวแปรชนิดโครงสร้าง ก็จำเป็นต้องประกาศตัวแปรให้ครบทุกตัว เช่น

```
#include<stdio.h>
int main()
{
char character1_name[20],character2_name[20],character3_name[20];
char character1_sname[20],character2_sname[20],character3_sname[20];
int  character1_age, character2_age, character3_age;
int  character1_b_date, character2_b_date, character3_b_date;
char character1_b_mounth[10],character2_b_mounth[10],character3_b_mounth[10];
char character1_gander[7],character2_gander[7],character3_gander[7];
int character1_height,character2_height,character3_height;
int character1_weight,character2_weight,character3_weight;
char character1_blood[4],character2_blood[4],character3_blood[4];
return 0;
}
```



- จากตัวอย่างที่ผ่านมา หากนำข้อมูลของตัวละครมาประกอบกันเป็นโครงสร้างของตัวละคร และใช้งาน **ตัวแปรชนิดโครงสร้าง** แทนการประกาศตัวแปรแบบปกติที่เรียนมา สามารถประกาศตัวแปรได้ดังนี้

```
#include<stdio.h>
int main()
{
    character {
        char name[20], sname[20];
        int age, b_date;
        char b_mounth[10], gander[7];
        int height, weight;
        char blood[4]
    } character_1, character_2, character_3;
    return 0;
}
```


2. โครงสร้างข้อมูล (Structure)

โครงสร้างข้อมูล (Structure) เป็นการกำหนดตัวแปรชนิดใหม่ขึ้นมาใช้งาน โดยนำตัวแปรพื้นฐานในภาษา C มารวมกันเป็นโครงสร้างของตัวแปรชนิดใหม่ เช่น

- โครงสร้างของตัวแปรนักเรียน ใช้ชื่อโครงสร้างเป็น **"student"**
ภายในโครงสร้างประกอบไปด้วย ข้อมูลชื่อ (name), นามสกุล (surname), อายุ (age), เพศ (gender) และ ห้อง (room)
- โครงสร้างของตัวแปรผลิตภัณฑ์ ใช้ชื่อโครงสร้างเป็น **"product"**
ภายในโครงสร้างประกอบไปด้วย ข้อมูลชื่อผลิตภัณฑ์ (name), จำนวน (count) และ ราคา (price)

โครงสร้างข้อมูล (Structure)

- วิธีทำความเข้าใจโครงสร้างข้อมูล ให้มองลักษณะภายในของตัวแปรชนิดโครงสร้างเป็นหัวของตาราง แล้วข้อมูลภายในตัวแปรชนิดโครงสร้างเป็นสมาชิกของตาราง

ชื่อโครงสร้าง

student

→ **name**
→ **surname**
→ **room**
→ **gender**
→ **age**



สมาชิกโครงสร้าง

student	name	surname	room	gender	age
#1	Gon	Freecss	1	male	12
#2	Killua	Zoaldyec k	2	male	12
#3	Kurapic a	Terminad a	9	male	17
.
.
#n

ตัวแปรชนิดโครงสร้าง

3. รูปแบบ การประกาศโครงสร้างข้อมูล

รูปแบบที่ 1

```
struct str_name {  
    type memvar_1;  
    type memvar_2;  
    .....  
    type memvar_n;  
};  
  
struct str_name struct_var;
```

ประกาศโครงสร้างข้อมูล แล้ว
สร้าง ตัวแปรชนิดโครงสร้าง ภายหลัง

รูปแบบที่ 2

```
struct str_name{  
    type memvar_1;  
    type memvar_2;  
    .....  
    type memvar_n;  
} struct_var;
```

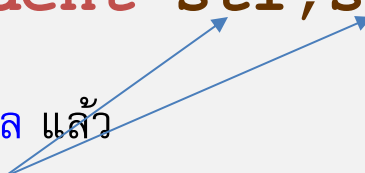
ประกาศโครงสร้างข้อมูล พร้อมกับ
สร้าง ตัวแปรชนิดโครงสร้าง

ตัวอย่าง การประกาศ ตัวแปรชนิดโครงสร้าง

รูปแบบที่ 1

```
struct student {  
    char name[15];  
    char surname[30];  
    int age;  
    char gender[7];  
};  
struct student st1, st2;
```

ประกาศโครงสร้างข้อมูล แล้ว
สร้าง ตัวแปรชนิดโครงสร้าง ภายหลัง



รูปแบบที่ 2

```
struct student {  
    char name[15];  
    char surname[30];  
    int age;  
    char gender[7];  
} st1, st2;
```

ประกาศโครงสร้างข้อมูล พร้อมกับสร้าง
ตัวแปรชนิดโครงสร้าง



4. การอ้างอิงข้อมูล ในตัวแปรชนิดโครงสร้าง

- การอ้างอิง หรืออ่านค่าของ ตัวแปรชนิดโครงสร้าง ทำได้โดยมีรูปแบบดังนี้

struct_var.member_var

struct_var คือ ชื่อตัวแปรชนิดโครงสร้าง

. ใช้ เชื่อมต่อระหว่างตัวแปรโครงสร้างและสมาชิก

member_var คือ ชื่อสมาชิกในตัวแปรชนิดโครงสร้าง

ตัวอย่าง การอ้างถึงข้อมูล ในตัวแปรชนิดโครงสร้าง

- สร้างโครงสร้างชื่อ product ซึ่งมีสมาชิกเป็น name, count, price

```
struct product {  
    char name[15];  
    int count;  
    float price;  
};  
struct product pr1, pr2;
```

- การอ้างถึงข้อมูลในตัวแปรชนิดโครงสร้าง เช่น

pr1.name pr2.count pr1.price

	name	count	price
pr1	pr1.name	pr1.count	pr1.price
pr2	pr2.name	pr2.count	pr2.price

5. การกำหนดค่าข้อมูลให้ ตัวแปรชนิดโครงสร้าง

- การกำหนดค่าข้อมูลให้กับตัวแปรชนิดโครงสร้าง ให้กำหนดค่าเหมือนตัวแปรปกติ โดยใช้วิธีอ้างอิงข้อมูล ของตัวแปรชนิดโครงสร้าง
- เช่น ต้องการกำหนด pr1.name เป็น Mouse มีค่า pr.count เป็น 99 ชิ้น และราคาต่อชิ้น เป็น 2190

	name	count	price
pr1	Mouse	99	2190
pr2			

~~pr1.name="Mouse";~~
pr1.count=99;
pr1.price=2190;

strcpy(pr1.name, "Mouse"); pr1.count=99; pr1.price=2190;

* **หมายเหตุ** คำสั่ง strcpy(); ต้องกำหนด พร้โปรเซสเซอร์ไคเร็คทีฟ #include<string.h>

ตัวอย่าง การกำหนดค่าข้อมูลให้ตัวแปรชนิดโครงสร้าง

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

```
int main()
```

```
{
```

```
    struct income{  
        float  salary;  
        float  bonus;  
        int   age;
```

```
    };
```

```
    struct income  people1;
```

```
    people1.salary = 16000;
```

```
    people1.bonus  = 40000;
```

```
    people1.age    = 23;
```

```
    return 0;
```

```
}
```

	salary	bonus	age
people1	16000	40000	23

ประกาศโครงสร้าง

ประกาศตัวแปรชนิดโครงสร้าง

กำหนดค่าข้อมูลให้ตัวแปรชนิดโครงสร้าง

ตัวอย่าง การรับข้อมูล และ กำหนดค่าให้แก่ตัวแปรชนิดโครงสร้าง

```
#include<stdio.h>
#include<string.h>
int main()
{
    struct letter{
        char name[20];
        char surname[30];
        char message[50];
    }first;
    //continue
```

```
        printf ("Enter name : ");
        scanf ("%s", first.name);
        printf ("Enter surname : ");
        scanf ("%s", first.surname);
        strcpy (first.message, "Hi");
        printf ("%s", first.message);
        printf ("\n%s", first.name);
        printf (" %s", first.surname);
        return 0;
    }
```

letter	name	surname	message
first	Jirasak	Sittigorn	Hi

```
(Inactive C:\TCWIN45\BIN\NONAME02.EXE)
Enter name : Jirasak
Enter surname : Sittigorn
Hi
Jirasak Sittigorn
```

ตัวอย่าง โปรแกรมรับข้อมูลหนังสือ 1 เล่ม

จงเขียนโปรแกรมสำหรับรับข้อมูลของหนังสือ 1 เล่ม โดยรับข้อมูล
ชื่อหนังสือ และ ราคาหนังสือ (โดยกำหนดใช้ตัวแปรชนิดโครงสร้าง
ภายในโครงสร้างกำหนดมีข้อมูล ชื่อหนังสือ, ราคา และส่วนลด
(คงที่ 10 %))

— ตัวอย่างการรันโปรแกรม

```
Enter book name   : Programming in TurboC  
Enter book price  : 200
```

```
Book   : Programming in TurboC  
Price  : 200.00  
Discount 10 percent : 20.00  
Total price           : 180.00
```

โปรแกรมรับข้อมูลหนังสือ 1 เล่ม

```
#include<stdio.h>
int main()
{
```

book	name	price	discount
book1	Programming in TurboC	200	20

```
    struct book
    {
        char name[50];
        float  price;
        float  discount;
    }book1;
```

```
    printf ("Enter book name  : ");
    gets    (book1.name);
    printf ("Enter book price : ");
    scanf   ("%f",&book1.price);
    book1.discount = 0.1*book1.price;
```

```
//continue
```

โปรแกรมรับข้อมูลหนังสือ 1 เล่ม

```
printf ("\n\n\n\nBook   : %s\n",book1.name) ;
printf ("Price : %.2f\n",book1.price) ;
printf ("Discount 10 percent : %.2f\n"
        "Total price           : %.2f",
        book1.discount,
        book1.price - book1.discount) ;

return 0;
}
```

Enter book name : Programming in TurboC

Enter book price : 200

Book : Programming in TurboC

Price : 200.00

Discount 10 percent : 20.00

Total price : 180.00

6. การกำหนดค่าเริ่มต้นให้แก่ โครงสร้าง

- การกำหนดค่าเริ่มต้นให้กับตัวแปรชนิดโครงสร้างคล้ายกับการกำหนดค่าเริ่มต้นให้แก่ตัวแปรทั่วไป แต่ ตัวแปรชนิดโครงสร้างจะต้องกำหนดค่าเริ่มต้นตามลำดับสมาชิกใน ตัวแปรโครงสร้างที่ประกาศไว้ เช่น

```
struct book
{
    char    name[50];
    float   price;
} b1 = {"harry", 120};
struct book b2 = {"Using C", 150};
struct book b3 = {150, "Pascal"};
```

	name char[50]	price float
b1	harry	120
b2	Using C	150
b3	150	Pascal

7. อาร์เรย์ของตัวแปรชนิดโครงสร้าง (Array of Structure)

- ในกรณีที่ต้องใช้ตัวแปรชนิดโครงสร้างที่เหมือนกันหลาย ๆ ตัว เราสามารถใช้คุณสมบัติของอาร์เรย์ กับโครงสร้างได้ เรียกว่า Array of Structure เช่น ต้องการตัวแปร student1, student2, ..., student10 จะสามารถกำหนดเป็นตัวแปรโครงสร้างเป็นอาร์เรย์ คือ **student[10]** เป็นต้น

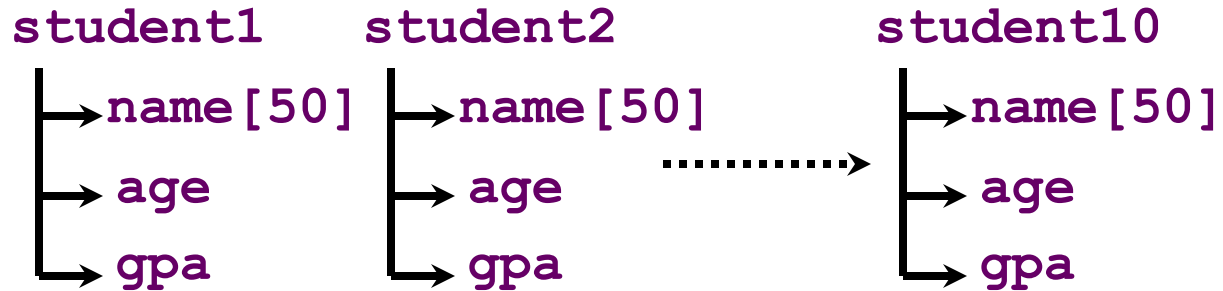
```
struct profile
{
    char   name[50];
    int    age;
    float  gpa;
};
```

```
struct profile student[10];
```

```
struct profile student1, student2, student3, ..., student10;
```

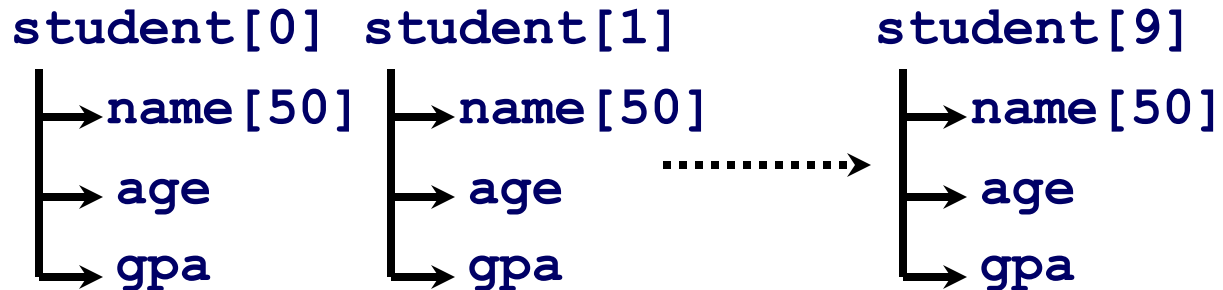
อาร์เรย์ของตัวแปรชนิดโครงสร้าง

```
struct profile student1, student2, student3, student4,  
               student5, student6, student7, student8,  
               student9, student10;
```



profile	name	age	gpa
student1			
student2			
...			
student10			

```
struct profile student[10];
```



profile	name	age	gpa
student[0]			
student[1]			
...			
student[9]			

ตัวอย่าง อาร์เรย์ของตัวแปรชนิดโครงสร้าง

```
#include<stdio.h>
int main()
{
    struct profile{
        char   name[20];
        int    age;
        char   grade;
    };
    struct profile student[10];
    return 0;
}
```

```
#include<stdio.h>
int main()
{
    struct profile{
        char   name[20];
        int    age;
        char   grade;
    } student[10];
    return 0;
}
```

profile	name	age	gpa
student[0]			
student[1]			
...			
student[9]			

โปรแกรมเก็บข้อมูลตัวละครจำนวน 10 คน

- จงเขียนโปรแกรมเก็บข้อมูลตัวละครจำนวน 10 คน โดยมีรายละเอียดดังนี้
 - ใช้ structure array เก็บข้อมูลตัวละคร
 - ข้อมูลตัวละครประกอบด้วย ชื่อ กับ อายุ
 - เมื่อป้อนข้อมูลตัวละครครบ โปรแกรมแสดงข้อมูลของตัวละครที่อายุน้อยกว่า 20 ปี

โปรแกรมเก็บข้อมูลตัวละครจำนวน 10 คน

— ตัวอย่างการรันโปรแกรม

Character 1

name : Gon

age : 12

Character 2

name : Kuroro

age : 26

Character 3

name : Hisoka

age : 27

Character 4

name : Killua

age : 12

Character 5

name : Machi

age : 24

Character 6

name : Nobunaga

age : 30

Character 7

name : Kurapica

age : 17

Character 8

name : Ubogin

age : 29

Character 9

name : Pakunoda

age : 28

Character 10

name : Leorio

age : 19

Gon,12

Killua,12

Kurapica,17

Leorio,19

โปรแกรมเก็บข้อมูลตัวละครจำนวน 10 คน

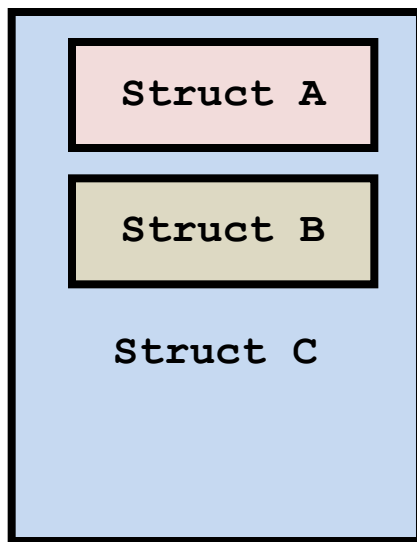
```
#include<stdio.h>
#define N 10
int main()
{
    int i;
    struct profile{
        char name[30];
        int age;
    }p[N];
    //continue
```

โปรแกรมเก็บข้อมูลตัวละครจำนวน 10 คน

```
for (i=0;i<N;i++)
{
    printf("Character %d\n",i+1);
    printf("\tname : ");
    scanf("%s",p[i].name);
    printf("\tage : ");
    scanf("%d",&p[i].age);
}
for (i=0;i<N;i++)
    if (p[i].age < 20)
        printf("\n %s,%d",p[i].name,p[i].age);
return 0;
}
```

8. โครงสร้างซ้อนโครงสร้าง (Nested Structure)

- เราสามารถสร้างโครงสร้างซ้อนโครงสร้างได้ เรียกว่า **Nest structure** เช่นประกาศโครงสร้าง **C** โดยภายในโครงสร้าง **C** ประกอบด้วยโครงสร้าง **A** และ **B** มีลักษณะดังรูป



```
struct A{  
    ...  
};  
struct B{  
    ...  
};  
struct C{  
    struct A data_a;  
    struct B data_b;  
    ...  
};
```

โครงสร้างซ้อนโครงสร้าง (Nested Structure)

```
struct address{
    int  num;
    int  moo;
    char road[20];
    char district[20];
    char province[20];
};
struct phone{
    char home[10];
    char mobile[10];
};
```

```
struct student{
    char  name[20];
    char  surname[20];
    char  id[9];
    struct address add;
    struct phone tel;
};
```

name	surname	id	add					tel	
			num	moo	road	district	province	home	mobile

ตัวอย่างโครงสร้างซ้อนโครงสร้าง

```
#include<stdio.h>
int main()
{
    struct address{
        int add;
        int moo;
        char road[20];
        char district[20];
        char province[20];
    };
    struct university{
        char name[70];
        struct address place;
    };
    //continue
```

ตัวอย่างโครงสร้างซ้อนโครงสร้าง

```
struct university king = {  
    "King Mongkut\'s Institute of Technology Lardkrabang",  
    {3,2,"Chalongkrung","Ladkraabang","Bangkok"}};  
printf ("%s\n",king.name);  
printf ("Address : %d Moo %d, %s Rd.\n",  
    king.place.add,king.place.moo,king.place.road);  
printf ("                %s, %s",king.place.district,  
    king.place.province);  
return 0;  
}
```

King Mongkut's Institute of Technology Lardkrabang
Address : 3 Moo 2, Chalongkrung Rd.
Ladkraabang, Bangkok


```
#include <stdio.h>
int main()
{
printf("Do you have any questions ?");
return 0;
}
```

ตัวอย่างแบบฝึกหัด

รับข้อมูล เป็นรายการหนังสือ โดยจะระบุ ID หนังสือ, ชื่อหนังสือ, ราคาหนังสือ, และชื่อผู้แต่งตามลำดับ จากนั้นให้ จะให้ระบุว่าต้องการซื้อหนังสืออะไร และจบการซื้อด้วยคำว่า End และจะแสดงผลสรุปว่า ราคาหนังสือที่ลดไป 20% จะเหลือราคาเล่มละกี่บาท ถ้าไม่มีการซื้อหนังสือใดๆ ให้แสดงผลว่า none

input	output
2 B0001 Math 50 Panwit B0002 PSIT 45 Chotipat Math End	B0001 Math 40.00 Baht
3 B0001 Compro 120 Panwit B0002 Multi 500 Supawan B0003 Comor 75 Anantapat Multi End	B0002 Multi 400.00 Baht