

บทที่ 10

ตัวแปรโครงสร้าง (Structure)





การส่งข้อมูลหลายค่าไปยังฟังก์ชัน

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

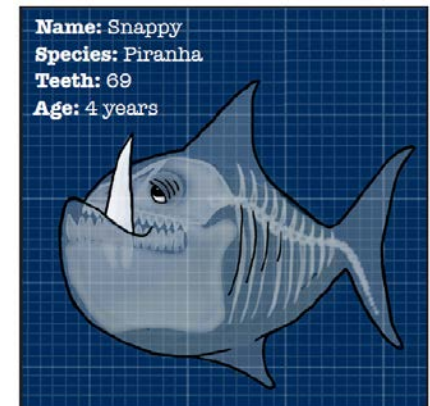
```
void catalog(char name[], char species[], int teeth, int age) {  
    printf("%s is a %s with %d teeth. He is %d\n\n", name,  
           species, teeth, age);  
}
```

```
void label(char name[], char species[], int teeth, int age) {  
    printf("Name:%s\nSpecies:%s\n%d years old, %d teeth\n", name,  
          species, teeth, age);  
}
```

```
void main() {  
    catalog("Snappy", "Piranha", 69, 4);  
    label("Snappy", "Piranha", 69, 4);  
}
```

ส่งข้อมูลปลาตัวเดียวกัน
ไปยัง 2 ฟังก์ชัน

มีปลาตัวเดียว แต่ต้องส่ง argument ไปหลายตัว





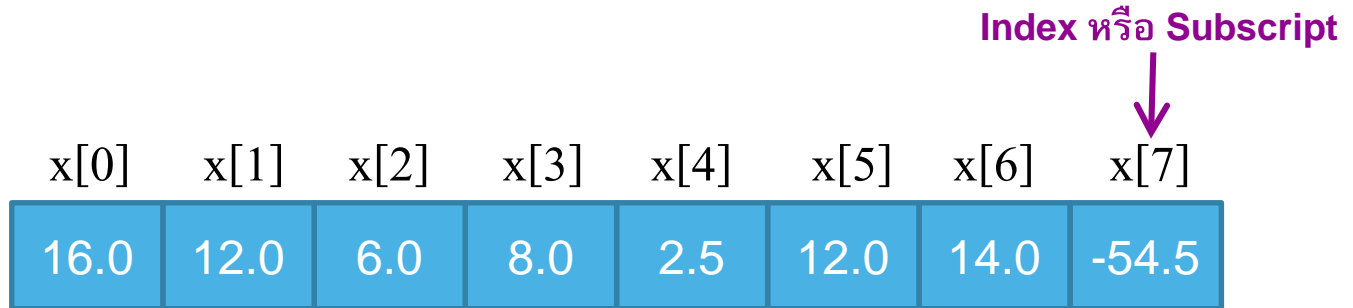
Structure

- ❖ Structure คือ ตัวแปรชนิด โครงสร้าง ที่สามารถเก็บมูลค่าได้หลายค่าเหมือนกับ array
- ❖ ตัวแปร Structure ต่างจาก array คือ สมาชิกทุกตัวไม่จำเป็นต้องเป็นชนิดข้อมูลเดียวกันก็ได้

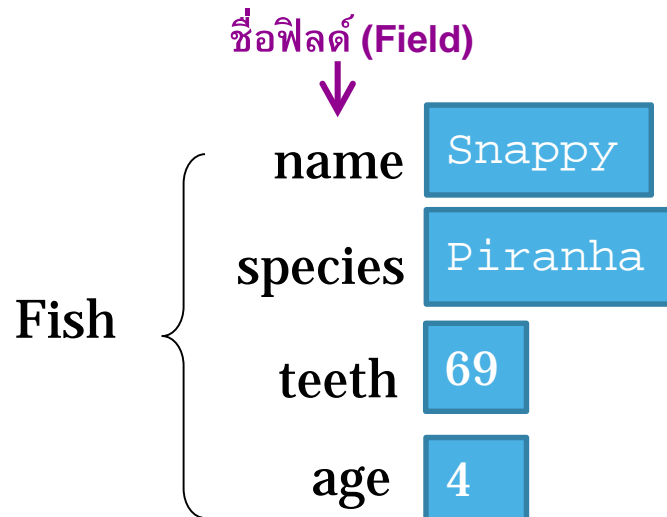


ตัวแปรชนิดโครงสร้าง

❖ อาร์เรย์



❖ ตัวแปรชนิดโครงสร้าง หรือ Structure





ทำไมต้องใช้ Structure

- ❖ การส่ง argument ที่มีข้อมูลเกี่ยวกับเรื่องเดียวกัน เช่น ข้อมูลนักศึกษา ข้อมูลสัตว์เลี้ยง ไปยัง function เมื่อมีการเปลี่ยนแปลงหรือเพิ่มตัวแปร จะต้องตามไปแก้ไขทุก function
- ❖ หากสามารถเก็บตัวแปรหลายชนิดเป็นก้อนเดียวกัน และส่งไปยังฟังก์ชันเหล่านี้ได้ จะง่ายต่อการพัฒนาโปรแกรมมากขึ้น



การสร้าง Structure

❖ รูปแบบ

```
typedef struct {  
    ชนิดข้อมูล ตัวแปรที่ 1;  
    ชนิดข้อมูล ตัวแปรที่ 2;  
    ...  
    ชนิดข้อมูล ตัวแปรที่ n;  
} ชื่อโครงสร้าง;
```

❖ ตัวอย่าง

```
typedef struct {  
    char name[20];  
    char species[10];  
    int teeth;  
    int age;  
} Fish;
```



กิจกรรม

จงออกแบบ Structure สำหรับเก็บข้อมูลรถยนต์



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



การประกาศตัวแปร Structure

- ❖ Structure ที่สร้างขึ้นเป็นเพียงโครงสร้างข้อมูลเท่านั้น ยังไม่ได้ถูกจองบนหน่วยความจำ การนำไปใช้จะต้องประกาศเหมือนกับตัวแปรตัวหนึ่ง ซึ่งมีรูปแบบดังนี้

ชื่อStructure ชื่อตัวแปร;

- ❖ ตัวอย่าง

Fish snappy;



การกำหนดค่าเริ่มต้น

❖ รูปแบบการประกาศค่าเริ่มต้นมีลักษณะเดียวกับอาร์เรย์ โดยจะกำหนดค่าของแต่ละฟิลด์ในเครื่องหมาย { }

```
ชื่อstructure ชื่อตัวแปร = { รายการค่าเริ่มต้นด้วย , };
```

```
Fish snappy = {"Snappy", "Piranha", 69, 4};
```



ตัวอย่าง

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

```
typedef struct {  
    char name[20];  
    char species[10];  
    int teeth;  
    int age;  
} Fish;
```

```
void main() {  
    Fish snappy = {"Snappy", "Piranha", 69, 4};  
    printf("%s\n", snappy.name);  
    printf("%s\n", snappy.species);  
    printf("%d\n", snappy.teeth);  
    printf("%d\n", snappy.age);  
}
```





กิจกรรม

จาก Structure สำหรับเก็บข้อมูลรถยนต์ในกิจกรรมก่อนหน้านี้ จงประกาศใช้ตัวแปร กำหนดค่าเริ่มต้น และพิมพ์ค่าในแต่ละฟิลด์ออกทางหน้าจอ



```
#include <stdio.h>
/* ละไว้ในส่วน Structure ที่ออกแบบไปแล้ว */
void main() {


---




---




---




---




---




---




---




---




---




---


}
```



การเข้าถึงและกำหนดค่าให้กับ Structure

❖ รูปแบบ

```
ชื่อstructure.ชื่อฟิลด์;
```

❖ ตัวอย่าง

```
snappy.age = 7;
```

```
scanf("%s", &snappy.name);
```

```
printf("%d\n", snappy.age);
```



ตัวอย่าง

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

```
typedef struct {  
    char name[20];  
    char species[10];  
    int teeth;  
    int age;  
} Fish;
```

```
void main() {  
    Fish snappy;  
    strcpy(snappy.name, "Snappy");  
    strcpy(snappy.species, "Piranha");  
    snappy.teeth = 69;  
    snappy.age = 4;  
    printf("%s %s %d %d\n", snappy.name, snappy.species,  
          snappy.teeth, snappy.age);  
}
```





การส่งตัวแปร structure ไปยังฟังก์ชัน

ส่งแบบหลายค่า

ส่งตัวแปรเดียว

```
#include <stdio.h>

void catalog(char name[], char species[], ...) {
    printf("...", name, species, teeth, age);
}

void label(char name[], char species[], ...) {
    printf("...", name, species, teeth, age);
}

void main() {
    catalog("Snappy", "Piranha", 69, 4);
    label("Snappy", "Piranha", 69, 4);
}
```

```
#include <stdio.h>

typedef struct {
    char name[20];
    char species[10];
    int teeth;
    int age;
} Fish;

void catalog(Fish f) {
    printf("...", f.name, f.species, f.teeth, f.age);
}

void label(Fish f) {
    printf("...", f.name, f.species, f.teeth, f.age);
}

void main() {
    Fish snappy = {"Snappy", "Piranha", 69, 4};
    catalog(snappy);
    label(snappy);
}
```



อาร์เรย์ของ Structure

❖ รูปแบบ

```
ชื่อStructure ชื่อตัวแปร[ขนาด];
```

❖ ตัวอย่าง

```
Fish fishList[4];
```




การกำหนดค่าเริ่มต้น

```
Fish fishList[4] = {  
    {"Snappy", "Piranha", 69, 4},  
    {"Golfy", "Tuna", 6, 1},  
    {"Nobita", "Shushi", 3, 2},  
    {"Pikachu", "Mackerel", 6, 15}  
};
```

	name	species	teeth	age
fishList[0]	Snappy	Piranha	69	4
fishList[1]	Golfy	Tuna	6	1
fishList[2]	Nobita	Shushi	3	2
fishList[3]	Pikachu	Mackerel	6	15



การกำหนดค่าและเข้าถึงสมาชิก

// การกำหนดค่า

```
strcpy(fishList[2].name, "Chinjang");  
fishList[2].age = 3;
```

// การรับค่า

```
scanf("%d", &fishList[2].teeth);
```

// การแสดงค่า

```
printf("%s\n", fishList[2].name);
```



ตัวอย่าง

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

```
typedef struct {  
    char name[20];  
    char species[10];  
    int teeth;  
    int age;  
} Fish;  
  
Fish inputFish() {  
    Fish f;  
    scanf("%s", &f.name);  
    scanf("%s", &f.species);  
    scanf("%d", &f.teeth);  
    scanf("%d", &f.age);  
    return f;  
}
```

```
void catalog(Fish f) {  
    printf("%s is a %s with %d teeth. He is %d\n\n",  
        f.name, f.species, f.teeth, f.age);  
}
```

```
void main() {  
    Fish fishList[4];  
  
    int i;  
    for(i=0; i<4; i++) {  
        fishList[i] = inputFish();  
    }  
  
    for(i=0; i<4; i++) {  
        catalog(fishList[i]);  
    }  
}
```



การใช้ pointer กับ structure

❖ รูปแบบ

ชื่อพอยเตอร์->ชื่อฟิลด์;

❖ ตัวอย่าง

```
Fish fisho;  
Fish *p;  
p = &fisho;  
scanf("%s", &p->name);  
scanf("%s", &p->species);
```



ตัวอย่าง

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

```
typedef struct {  
    char name[20];  
    char species[10];  
    int teeth;  
    int age;  
} Fish;
```

```
void catalog(Fish f) {  
    printf("%s is a %s with %d teeth. He is %d\n\n",  
        f.name, f.species, f.teeth, f.age);  
}
```

```
void main() {  
    Fish fisho;  
    Fish *p;  
    p = &fisho;  
  
    scanf("%s", &p->name);  
    scanf("%s", &p->species);  
    scanf("%d", &p->teeth);  
    scanf("%d", &p->age);  
  
    catalog(fisho);  
}
```



กิจกรรม

จงเขียนโปรแกรมคำนวณระยะห่างระหว่างจุด 2 จุด คือ $P1(x_1, y_1)$ และ $P2(x_2, y_2)$ โดยใช้สูตร $distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

โดยสร้าง Structure ชื่อ Point เพื่อใช้ในการเก็บพิกัด และเขียนฟังก์ชันดังนี้

1. ฟังก์ชันรับข้อมูลทางคีย์บอร์ด โดยเก็บข้อมูลพิกัดลงตัวแปร Structure

```
Point getPoint();
```

2. ฟังก์ชันหาค่าระยะทางระหว่างจุด 2 จุด

```
float distance(Point p1, Point p2);
```

ตัวอย่างหน้าจอโปรแกรม

```
Enter P1 (x1,y1) (separate by space): 5 6 ↵  
Enter P2 (x2,y2) (separate by space): 2 3 ↵  
distance = 4.24
```



Assignment#10

ข้อ 1. จงเขียนโปรแกรมรับข้อมูลรหัส
นักศึกษา ชื่อนักศึกษา และคะแนน 4 คน
โดยออกแบบ Structure และสร้างเป็น
อาร์เรย์ของ Structure เก็บข้อมูลที่รับจากผู้ใ้
หลังจากรับค่าเสร็จให้ดึงค่าออกมาแสดง
พร้อมกับหาค่าเฉลี่ยของคะแนน

```
Student ID: 562030111-1 ↵  
Student Name: Jim ↵  
Score: 55 ↵
```

```
Student ID: 562030111-2 ↵  
Student Name: John ↵  
Score: 76 ↵
```

```
Student ID: 562030111-3 ↵  
Student Name: Joe ↵  
Score: 49 ↵
```

```
Student ID: 562030111-4 ↵  
Student Name: Jimmy ↵  
Score: 76 ↵
```

ID	Name	Score
562030111-1	Jim	55
562030111-2	John	76
562030111-3	Joe	49
562030111-4	Jimmy	76

Score Average = 64.00

ตัวอย่างการทำงาน



Assignment#10

ข้อ 2. จงเขียนโปรแกรมรับคะแนนสอบวิชาการเขียน โปรแกรมภาษาซีของนักเรียน n คน เพื่อคำนวณหาเกรดที่จะได้ โดยแบ่งคะแนนออกเป็น คะแนนสอบ Lab 15 คะแนน คะแนนสอบ Midterm และ Final 30 คะแนน คะแนนเข้าเรียน 10 คะแนน และคะแนนงาน 15 คะแนน ให้ออกแบบ Structure และสร้างเป็นอาร์เรย์ของ Structure เก็บข้อมูลที่รับจากผู้ใช้ หลังจากรับค่าเสร็จให้ดึงค่าออกมาแสดง พร้อมกับแสดงเกรดที่ได้ ซึ่งมีเกณฑ์ดังนี้

ช่วงคะแนน	เกรดที่ได้
คะแนนมากกว่าหรือเท่ากับ 80	A
คะแนนมากกว่าหรือเท่ากับ 70	B
คะแนนมากกว่าหรือเท่ากับ 60	C
คะแนนมากกว่าหรือเท่ากับ 50	D
คะแนนน้อยกว่า 50	F



Assignment#10

ตัวอย่างการทำงาน

Enter amount : 3

Enter Student no.1

```
>>> Enter student id : 1 ↵
>>> Enter score Laboratory Test : 14 ↵
>>> Enter score Midterm Test : 23 ↵
>>> Enter score Homework : 11 ↵
>>> Enter score Final Test : 20 ↵
```

Enter Student no.2

```
>>> Enter student id : 2 ↵
>>> Enter score Laboratory Test : 2 ↵
>>> Enter score Midterm Test : 10 ↵
>>> Enter score Homework : 12 ↵
>>> Enter score Final Test : 11 ↵
```

Enter Student no.3

```
>>> Enter student id : 3 ↵
>>> Enter score Laboratory Test : 1 ↵
>>> Enter score Midterm Test : 5 ↵
>>> Enter score Homework : 15 ↵
>>> Enter score Final Test : 10 ↵
```

ID	Lab Test	Midterm	Homework	Final	Sum	Grade
1	14.00	23.00	11.00	20.00	68.00	C
2	2.00	10.00	12.00	11.00	35.00	F
3	1.00	5.00	15.00	10.00	31.00	F