

Structure

NATSUDA KAOTHANTHONG

CS102 (NOVEMBER 2015)

Last Lecture: 2D Array (อะเรย์ 2 มิติ หรือ อะเรย์ของอะเรย์ 1 มิติ)

`int num[3][5];`

จำนวนอะเรย์หลัก (row)

จำนวนอะเรย์ย่อย (column)

- การเรียกใช้ข้อมูลในแต่ละช่อง ทำได้โดยใช้ **index** โดยเริ่มจาก **[row]** และตามด้วย **[column]**

num

[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]

This Lecture

- ❖ การกำหนดชนิดข้อมูลโดยใช้ **structure**
- ❖ การประกาศ **structure**
- ❖ **struct** และ **function**
- ❖ สมาชิกภายใน **struct** ที่เป็น พอยน์เตอร์ และ อาเรย์
- ❖ การกำหนดค่าและการเรียกใช้ข้อมูลที่อยู่ภายใน **structure**
- ❖ การกำหนดข้อมูลชุดของ **struct** (Array of struct)

Limitation of Primitive Type - 1

- ❖ **Primitive type variable** คือชนิดของตัวแปรที่กำหนดไว้ในภาษาซี
 - ❖ `int`, `float`, `double`, `char`, `long int`, etc.

ตัวอย่าง ตัวแปรสำหรับเก็บข้อมูลของสินค้า

`char name_A[100];` //สำหรับเก็บชื่อสินค้าA

`float price_A;` //สำหรับเก็บราคาสินค้าA

`int amount_A;` //จำนวนสินค้าAที่มีอยู่

- สำหรับสินค้า 1 ชิ้น ต้องใช้ตัวแปร 3 ตัว
- หากต้องการเก็บข้อมูลสินค้า 5 ตัว ต้องใช้ตัวแปรทั้งหมด 15 ตัว

Limitation of Primitive Type - 2

❖ ปัญหา1

- ❖ ข้อมูลของสินค้าแต่ละตัวอยู่กระจัดกระจาย และจำนวนตัวแปรที่ต้องใช้มีเยอะจนอาจทำให้สับสนเมื่อต้องการใช้ข้อมูล

❖ ปัญหา2

- ❖ ชนิดของข้อมูลของสินค้า1ชิ้น มีชนิดต่างกัน
 - ❖ นั่นคือ char, float, และ int
- ❖ ดังนั้น ไม่สามารถจัดเก็บข้อมูลโดยใช้อะเรย์ (Array)
 - ❖ char pro_names[50][100];
 - ❖ float pro_price[50];

ข้อมูลในอะเรย์ต้องมีชนิดเดียวกันเท่านั้น

This Lecture

- ❖ การกำหนดชนิดข้อมูลโดยใช้ structure
- ❖ การประกาศ structure
- ❖ struct และ function
- ❖ สมาชิกภายในstructที่เป็น พอยน์เตอร์ และ อาเรย์
- ❖ การกำหนดค่าและการเรียกใช้ข้อมูลที่อยู่ใน structure
- ❖ การกำหนดข้อมูลชุดของ struct (Array of struct)

Structure – struct

- ❖ ในภาษาซี เราสามารถสร้างชนิดตัวแปรได้เองโดยใช้ structure

struct ชื่อเรียก

```
{
    ประกาศตัวแปรต่าง ๆ
    (member)
    Primitive type
};
```

struct **product**

```
{
    char name[100]; //ชื่อสินค้า
    float price; //ราคาสินค้า
    int amount; //จำนวนสินค้า
};
```

- ❖ struct ชื่อเรียก ใช้เป็นชนิดข้อมูลที่สร้างขึ้น (struct **product**)

- ❖ ตัวแปรต่าง ๆ (member) คือ รายการสมาชิกที่อยู่ในชนิดข้อมูลใหม่ที่เราสร้างขึ้น

การประกาศตัวแปรที่ใช้ชนิดที่สร้างขึ้น

```
struct product
{
    char name[100]; //ชื่อสินค้า
    float price; //ราคาสินค้า
    int amount; //จำนวนสินค้า
};
```

- ❖ การประกาศตัวแปรโดยใช้ ชนิดที่สร้างขึ้น

struct **product** myStore;
ชื่อของชนิดที่สร้างขึ้น ชื่อตัวแปร

- ❖ หากต้องการประกาศตัวแปร พร้อมกำหนดค่าเริ่มต้นให้สมาชิกแต่ละตัว

struct product myStore = {"KitKat", 20.0, 155};

- ❖ ลำดับของค่าที่กำหนดภายใน { } ต้องเป็นลำดับเดียวกับที่ประกาศชนิดไว้
- ❖ ข้อควรระวัง

✗ struct product myStore;
myStore = {"KitKat", 20.0, 155};

ประกาศ และ กำหนดค่า
แบบนี้ไม่ได้

การกำหนดค่าให้ตัวแปรแต่ละตัวใน struct

```
struct product
```

```
{  
    char name[100]; //ชื่อสินค้า  
    float price; //ราคาสินค้า  
    int amount; //จำนวนสินค้า  
};
```

```
struct product myStore;
```

ชื่อของชนิดที่สร้างขึ้น ชื่อตัวแปร

การเรียกใช้สมาชิกแต่ละตัวที่อยู่ใน struct

ชื่อตัวแปร.ชื่อสมาชิก

ตัวอย่าง `strcpy(myStore.name, "Kewpie");`
 `myStore.price = 85.50;`
 `scanf("%d", &(myStore.amount));`

ตัวอย่าง struct

สามารถประกาศ struct ในฟังก์ชัน main หรือ นอกฟังก์ชัน main ก็ได้

```
int main()
```

```
{  
    struct studentRecord  
    {  
        char name[100];  
        char lastname[100];  
        float gpa;  
    }  
    struct studentRecord std;  
}
```

ถ้าประกาศ struct studentRecord ในฟังก์ชัน main ฟังก์ชันอื่น ๆ จะไม่รู้จัก struct studentRecord

```
struct studentRecord
```

```
{  
    char name[100];  
    char lastname[100];  
    float gpa;  
}  
int main()  
{  
    struct studentRecord std;  
}
```

ถ้าประกาศ struct studentRecord ภายนอกฟังก์ชัน main ฟังก์ชันอื่น ๆ สามารถประกาศตัวแปรชนิด struct studentRecord ได้

การกำหนดชื่อเรียก struct ที่สร้างขึ้น

กำหนดชื่อให้กับชนิดที่เป็น struct ที่สร้างขึ้นโดยใช้คำสั่ง **typedef**

❖ การใช้ typedef ทำได้ 2 วิธี

1. โดยใช้ **typedef struct** ตามด้วย **ชื่อที่ต้องการกำหนด**

```
struct product
```

```
{  
    char name[100]; //ชื่อสินค้า  
    float price; //ราคาสินค้า  
    int amount; //จำนวนสินค้า  
};
```

```
typedef struct product prod
```

ชื่อเรียกที่กำหนด

2. โดยใช้ **typedef** เมื่อประกาศ **struct**

```
typedef struct
```

```
{  
    char name[100]; //ชื่อสินค้า  
    float price; //ราคาสินค้า  
    int amount; //จำนวนสินค้า  
}  
prod;
```

ชื่อเรียกที่กำหนด

```
prod myStore1, myStore2, myStore3;
```

ชื่อของชนิดที่กำหนด

ชื่อตัวแปร

การอ้างอิงค่าจากตัวแปร struct

```
bookInfo book = {"Intro to C", "Perry", 1432};
```

```
typedef struct
```

```
{  
    char name[100];  
    char author[100];  
    int isbn;  
}bookInfo;
```

การเรียกใช้ค่าจาก book

- `printf("%s", book.name);`

- `printf("%d", book.isbn);`

ชื่อตัวแปรที่เป็นชนิด struct. ชื่อสมาชิก

- `scanf("%d", &(book.isbn));`

การอ้างอิงค่าจากตัวแปร struct ที่เป็นพอยน์เตอร์

```
typedef struct
{
    char name[100];
    char author[100];
    int isbn;
}bookInfo;
```

```
bookInfo book = {"Intro to C", "Perry", 1432};
```

```
bookInfo *bookPtr = &book;
```

การอ้างอิงค่าจาก ***bookPtr** ทำได้ 2 วิธี

1. `printf("%s", (*bookPtr).name);`
(***structที่เป็นpointer**).ชื่อสมาชิก
2. `printf("%s", bookPtr->name);`
structที่เป็นpointer->ชื่อสมาชิก

This Lecture

- ❖ การกำหนดชนิดข้อมูลโดยใช้ structure
- ❖ การประกาศ structure
- ❖ struct และ function
- ❖ สมาชิกภายในstructที่เป็น พอยน์เตอร์ และ อาร์เรย์
- ❖ การกำหนดค่าและการเรียกใช้ข้อมูลที่อยู่ภายใน structure
- ❖ การกำหนดข้อมูลชุดของ struct (Array of struct)

Struct and Function

การส่งข้อมูลที่เป็น struct ไปยังฟังก์ชันอื่น ๆ ทำได้ 2 วิธี

1. Pass-by-value

```
typedef struct
{
    char name[100];
    float price;
    int amount;
}prod;
void printProduct(prod store)
{
    printf("Product: %s, Price %f,
        Amount %d\n", store.name,
        store.price, store.amount);
}
int main()
{
    prod myStore1={"Tsubaki", 299, 5};
    printProduct(myStore1);
    return 0;
}
```

```
typedef struct
{
    char name[100];
    float price;
    int amount;
}prod;
void enterPrice(prod *storePtr)
{
    printf("Enter price of %s :", (*storePtr).name);
    scanf("%f", &((*storePtr).price));
}
int main()
{
    prod myStore1={"Tsubaki", 0, 5};
    enterPrice (&myStore1);
    return 0;
}
```

(***storePtr**).name และ **storePtr->name** มีความหมายเหมือนกัน
ใช้ **&(storePtr->price)** เมื่อต้องการที่อยู่ของสมาชิกที่ประกาศอยู่ภายใน
ตัวแปรstructที่เป็นพอยน์เตอร์

Returning struct from function

การreturnค่าที่เป็น struct จากฟังก์ชันหนึ่งไปอีกฟังก์ชันหนึ่ง

```
typedef struct
{
    char name[100];
    float price;
    int amount;
}prod;
prod newProduct(void)
{
    prod addProduct;
    printf("Enter name, price, and amount of the new product:");
    scanf("%s %f %d", &addProduct.name, &addProduct.price, &addProduct.amount);
    return addProduct;
}
int main()
{
    prod myNewProduct;
    myNewProduct = newProduct();
    return 0;
}
```

Exercise

- กำหนด **structure** ให้ดังต่อไปนี้ เขียนโปรแกรมรับหนังสือ 2 เล่มจากผู้ใช้

```
typedef struct
{
    char name[100];
    char author[100];
    int isbn;
}bookInfo;
```

- เขียนฟังก์ชัน `bookInfo editAuthor(bookInfo book, char name[100])` สำหรับแก้ไขชื่อผู้แต่ง
- เขียนฟังก์ชัน `void editAuthor_ref(bookInfo *book, char name[100])` สำหรับแก้ไขชื่อผู้แต่งโดยใช้ `call-by-reference`
- เขียนฟังก์ชัน `int compare(bookInfo b1, bookInfo b2)` เพื่อเปรียบเทียบว่าหนังสือทั้ง 2 เล่มมีเลข ISBN ที่ตรงกันหรือไม่ ถ้าตรงกันให้คืนค่า 1 มิฉะนั้น 0

This Lecture

- ❖ การกำหนดชนิดข้อมูลโดยใช้ **structure**
- ❖ การประกาศ **structure**
- ❖ **struct** และ **function**
- ❖ สมาชิกภายใน **struct** ที่เป็น พอยน์เตอร์ และ อาเรย์
- ❖ การกำหนดค่าและการเรียกใช้ข้อมูลที่อยู่ภายใน **structure**
- ❖ การกำหนดข้อมูลชุดของ **struct** (Array of struct)

Pointer Variable - Member

- ❖ ภายใน **struct** สามารถกำหนดชนิดสมาชิกที่เป็น **Pointer** ได้

```
กำหนด struct ดังต่อไปนี้
typedef struct
{
    char name[100];
    char author[100];
    int isbn;
    int *typeld;
}bookInfo;
```

```
int type[]={101, 102, 103, 104, 105};
bookInfo book1 = {"C Beginner", "Perry", 11543, &type[3]};
❖ การเข้าถึงค่าของสมาชิกที่เป็น pointer
printf("Book type ID is %d", *(book1.typeld));
```

- ❖ ข้อควรระวัง

- ❖ กำหนด `bookInfo *bookPtr = &book1;`

- ❖ `(*bookPtr)`.ชื่อสมาชิก

ใช้สำหรับอ้างถึงตัวแปรชนิด `bookInfo` ที่เป็น `pointer`

- ❖ `*(book1.ชื่อสมาชิก)`

ใช้สำหรับอ้างถึงสมาชิกภายใน **structure** ที่เป็น `pointer` เช่น `*(book1.typeld)`

Pointer Variable - Member

```
int type[]={101, 102, 103, 104, 105};
```

```
bookInfo book1 = {"C Beginner", "Perry", 11543, &type[3]};
```

```
bookInfo book2 = book1;
```

- ❖ ข้อควรระวัง

- ❖ ค่าของตัวแปร `pointer` ถูกคัดลอกมาโดยอัตโนมัติ

- ❖ ค่าของ `*(book1.typeld)` และ `*(book2.typeld)` มีค่าเท่ากัน

- ❖ ดังนั้น การเปลี่ยนค่าจาก `*(book2.typeld)` มีผลกับ `*(book1.typeld)`

Array - Member

กำหนด struct ดังต่อไปนี้

```
typedef struct
{
    char name[100];
    char author[100];
    int isbn;
    float margin[4];
}bookInfo;
```

ตัวอย่าง ตัวแปร Array ที่เป็นสมาชิกอยู่ใน structure

❖ ตัวแปร `margin[4]` สำหรับเก็บค่าความกว้างของหัวกระดาษ

❖ การกำหนดค่าทำได้ 2 แบบ

1. `bookInfo book;`

❖ `book.margin[0] = 2.3;`

❖ `book.margin[1] = 2.1;`

2. `bookInfo book2 = {"Intro to C", "Perry", 1432, {2.1, 2.3, 1.4, 1.6}};`

❖ การเรียกใช้ค่า

❖ `printf("Left margin : %.2f", book2.margin[0]);`

This Lecture

❖ การกำหนดชนิดข้อมูลโดยใช้ structure

❖ การประกาศ structure

❖ struct และ function

❖ การกำหนดค่าและการเรียกใช้ข้อมูลที่อยู่ภายใน structure

❖ การกำหนดข้อมูลชุดของ struct (Array of struct)

Struct Variable in Another Struct

❖ การกำหนดตัวแปรที่เป็น struct ไว้ใน อีกstructureหนึ่ง

กำหนด struct ดังต่อไปนี้

```
typedef struct
{
    char name[100];
    authorInfo author;
    int isbn;
}bookInfo;
```

กำหนดอีก struct ดังต่อไปนี้

```
typedef struct
{
    char name[100];
    char dob[50];
}authorInfo;
```

การกำหนดค่าเริ่มต้น

```
bookInfo book = {"Intro to C", {"Perry", "22/09/80"}, 1543};
```

struct ของ authorInfo

การเข้าถึงสมาชิกที่เป็น structure ภายใน structure

```
typedef struct
{
    char name[100];
    authorInfo author;
    int isbn;
}bookInfo;
```

❖ `bookInfo book;`

❖ การเข้าถึงสมาชิกที่เป็น structure

ตัวแปร `structure`.ชื่อสมาชิก `structure`.ชื่อสมาชิก

❖ `strcpy(book.author.name, "John");`

```
typedef struct
{
    char name[100];
    char dob[50];
}authorInfo;
```

This Lecture

- ❖ การกำหนดชนิดข้อมูลโดยใช้ structure
- ❖ การประกาศ structure
- ❖ การกำหนดค่าและการเรียกใช้ข้อมูลที่อยู่ใน structure
- ❖ struct และ function
- ❖ การกำหนดข้อมูลชุดของ struct (Array of struct)

Array of Struct

กำหนด struct ดังต่อไปนี้

```
typedef struct
{
    char name[100]; //ชื่อสินค้า
    float price; //ราคาสินค้า
    int amount; //จำนวนสินค้า
}prod;
```

ประกาศอะเรย์ของ struct

```
prod allProduct[100];
```

- ❖ คล้ายกับการประกาศอะเรย์ให้กับตัวแปรทั่วไป (Primitive Type)
 - ❖ เช่น int num[20]; float decimalPoint[30]; และ char firstName[50];
- ❖ เว้นแต่ชนิดตัวแปรที่อยู่ในเป็น struct ซึ่งมีสมาชิกที่เป็นตัวแปรชนิดต่าง ๆ อยู่

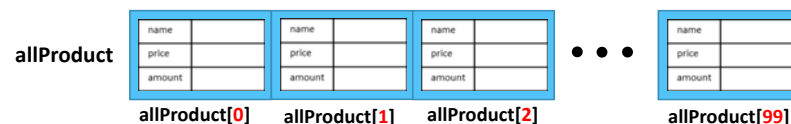


การกำหนดค่าให้สมาชิกแต่ละตัวในอะเรย์

ประกาศอะเรย์ของ struct

```
prod allProduct[100];
```

- ❖ ใช้ `scanf()`
- ```
scanf("%s %f %d", allProduct[0].name, &(allProduct[0].price), &(allProduct[0].amount));
```
- ❖ กำหนดค่าให้แต่ละสมาชิกในแต่ละ index ของอะเรย์
- ```
strcpy(allProduct[2].name, "Linn");
allProduct[2].price = 20;
allProduct[2].amount = 2;
```



Array of Struct and Function (call-by-value)

1. Pass-by-value

```
typedef struct
{
    char name[100];
    float price;
    int amount;
}prod;

prod newProduct(void)
{
    prod addProduct;
    printf("Enter name, price, and amount of the new product:");
    scanf("%s %f %d", addProduct.name, &(addProduct.price), &(addProduct.amount));
    return addProduct;
}

int main()
{
    prod allProduct[100];
    int i;
    for(i = 0; i < 100; i++)
    {
        allProduct[i] = newProduct();
    }
    return 0;
}
```

ประกาศ struct

ใช้รูปแบบการกำหนดค่าให้ struct แต่ละตัวในอะเรย์

Array of Struct and Function (call-by-reference)

2. Pass-by-reference

```
typedef struct
{
    char name[100];
    float price;
    int amount;
}prod;

void newProduct(prod *storePtr)
{
    printf("Enter name, price, and amount of the new product:");
    scanf("%s %f %d",(*storePtr).name, &((*storePtr).price), &((*storePtr). amount) );
}

int main()
{
    prod allProduct[100];
    int i;
    for(i = 0; i < 100; i++)
    {
        newProduct( &allProduct[i] );
    }
    return 0;
}
```

ประกาศ struct

ใช้รูปเพื่อกำหนดค่าให้ struct แต่ละตัวในอะเรย์โดยใช้ pointer

Exercise

กำหนดอะเรย์ให้ดังต่อไปนี้

```
typedef struct
{
    char name[100];
    char author[100];
    int isbn;
}bookInfo;
```

- เขียนโปรแกรมรับข้อมูลของหนังสือ 5 เล่มโดยใช้ call-by-reference
- เขียนฟังก์ชัน 3 ฟังก์ชันดังต่อไปนี้
 - searchBook เพื่อให้ผู้ใช้สามารถค้นหาหนังสือโดยใช้ชื่อหนังสือ
 - searchAuthor เพื่อให้ผู้ใช้สามารถค้นหาหนังสือโดยใช้ชื่อผู้แต่ง
 - searchISBN เพื่อให้ผู้ใช้สามารถค้นหาหนังสือโดยใช้ รหัส ISBN