



CS311: Computer Programming II

Week3: Array of Structure in C

Website: http://bit.ly/sirinthorn_c

A.Sirinthorn Cheyasak

E-mail: sirinthorn.c@bu.ac.th

School of Science and Technology, Bangkok University

วัตถุประสงค์การเรียนรู้

- เพื่อให้รู้จักตัวแปรชุดโครงสร้างและเข้าใจถึงรูปแบบการจัดเก็บข้อมูลในตัวแปรชุดโครงสร้าง
- เพื่อศึกษาและสามารถนิยามโครงสร้างพร้อมสามารถนำโครงสร้างไปประยุกต์ใช้สำหรับเก็บข้อมูลเพื่อนำไปประมวลผลได้
- เพื่อให้สามารถส่งผ่านค่าของโครงสร้างไปยังฟังก์ชันด้วยพารามิเตอร์
- สามารถนิยามโครงสร้างซ้อนโครงสร้าง พร้อมสามารถนำโครงสร้างไปประยุกต์สำหรับการจัดเก็บข้อมูลได้



รายละเอียดเนื้อหา (Outline)

- การประกาศตัวแปรชุดโครงสร้าง
- การเข้าถึงตัวแปรชุดโครงสร้าง
- การเขียนโปรแกรมประยุกต์สำหรับตัวแปรชุดโครงสร้าง
- การประกาศโครงสร้างซ้อนโครงสร้าง
- การเข้าถึงตัวแปรโครงสร้างซ้อนโครงสร้าง
- การเขียนโปรแกรมประยุกต์สำหรับตัวแปรโครงสร้างซ้อนโครงสร้าง



ทำไมจึงต้องเก็บข้อมูลแบบชุดโครงสร้าง

- ข้อมูลแบบชุดโครงสร้าง (Array of Structures) สามารถนำมาประยุกต์ใช้เพื่อรองรับข้อมูลปัจจุบันมีความหลากหลายและมีจำนวนมากขึ้น เช่น ข้อมูลนักเรียน ข้อมูลลูกค้า ข้อมูลสินค้า เช่น
 - ข้อมูลนักเรียนประกอบด้วย รหัสนักเรียน ชื่อสกุล อายุ เพศ ระดับชั้นเกรด เป็นต้น
 - ข้อมูลลูกค้าประกอบด้วย รหัสลูกค้า ชื่อสกุล ที่อยู่ อายุ ประวัติการซื้อสินค้า เป็นต้น
 - ข้อมูลสินค้าประกอบด้วย รหัสสินค้า ชื่อสินค้า ประเภทสินค้า ราคาสินค้า จำนวนสินค้าในคลัง เป็นต้น



ทำไมจึงต้องเก็บข้อมูลแบบชุดโครงสร้าง (ต่อ)

- การเก็บข้อมูลสำหรับการประมวลผล
 - โดยส่วนมากการเก็บข้อมูลจะมีข้อมูลมากกว่า 1 ชุดขึ้นไป เช่น การเก็บข้อมูลนักเรียนห้องหนึ่งซึ่งมีจำนวนนักเรียน 30 คน ดังนั้นเราสามารถสร้างตัวแปรของนักเรียนให้เป็นตัวแปรชุดโครงสร้าง เพื่อทำการเก็บข้อมูลแบบโครงสร้างไว้เป็นกลุ่มข้อมูลที่เกี่ยวข้งกัน เช่น โครงสร้างนักเรียน ซึ่งโรงเรียนหนึ่งจะต้องมีนักเรียนหลายคนจะต้องเก็บข้อมูลของนักเรียนหลายๆ คนเพื่อนำมาประมวลผลต่อไป



การประกาศโครงสร้างเพื่อเก็บข้อมูลด้วย struct

- การประกาศข้อมูลโครงสร้างข้อมูลด้วย typedef มีรูปแบบดังนี้

รูปแบบ	<pre>typedef struct { ชนิดข้อมูล ชื่อฟิลด์ข้อมูลที่ 1; ชนิดข้อมูล ชื่อฟิลด์ข้อมูลที่ 2; ... ชนิดข้อมูล ชื่อฟิลด์ข้อมูลที่ n; } ชื่อโครงสร้างข้อมูล ;</pre>
--------	---

ชื่อโครงสร้างข้อมูล ชื่อตัวแปรชุดโครงสร้าง[ขนาดข้อมูล];



ตัวอย่างการประกาศตัวแปรชุดโครงสร้าง

- ตัวอย่างการประกาศโครงสร้าง พร้อมตัวแปรชุดโครงสร้างดังนี้

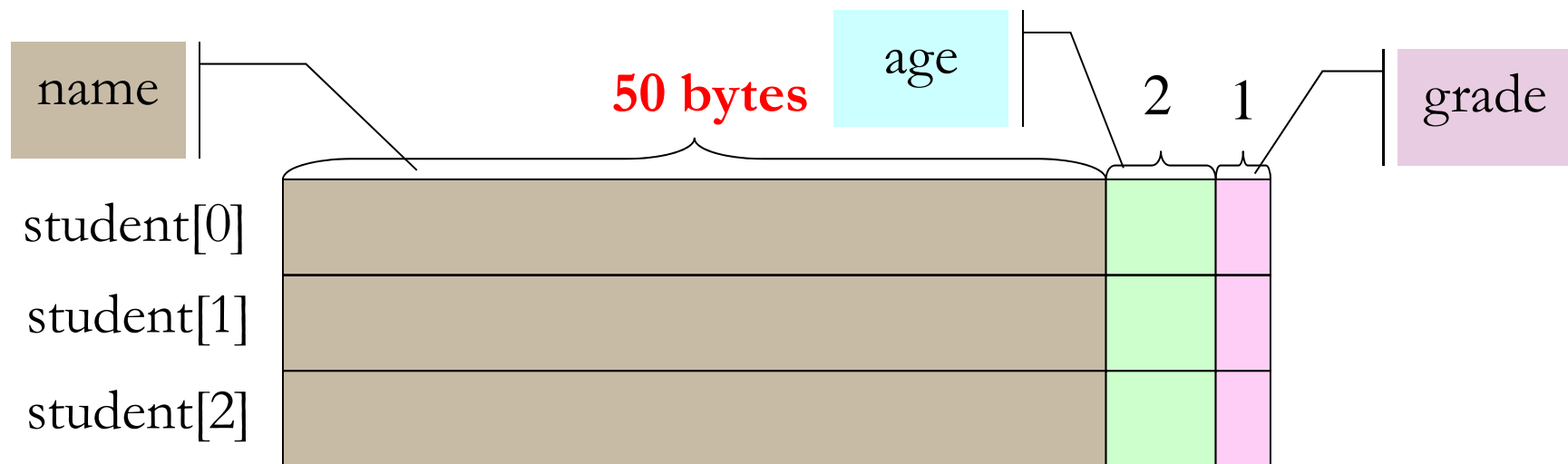
```
typedef struct {  
    char name[50];  
    int age;  
    char grade;  
} profile;  
int main()  
{  
    profile student[30];  
}
```

- ตัวแปรชุดโครงสร้างชื่อ student ที่มีขนาดข้อมูล 30 ช่อง (เพื่อเก็บข้อมูลจำนวนนักเรียน 30 คน)
- ข้อมูลในแต่ละช่องข้อมูลมีโครงสร้างการเก็บข้อมูลแบบ profile ซึ่งประกอบด้วย
 - ชื่อขนาด 50 ตัวอักษร
 - อายุ
 - เกรด



ตัวอย่างภาพโครงสร้างการเก็บข้อมูลตัวแปรชุดโครงสร้าง

```
typedef struct {  
    char name[50];  
    int  age;  
    char grade;  
} profile;
```



การกำหนดค่าเริ่มต้นให้กับตัวแปรชุดโครงสร้าง

- การกำหนดค่าเริ่มต้นของตัวแปรชุดโครงสร้างจะต้องกำหนดค่าเริ่มต้นให้ตรงกับสมาชิกหรือฟิลด์ของโครงสร้างนั้นๆ

รูปแบบ	ชื่อโครงสร้าง ชื่อตัวแปรโครงสร้าง[ขนาดข้อมูล] = { { รายการสมาชิกของตัวแปรชุดตัวที่ 1 }, { รายการสมาชิกของตัวแปรชุดตัวที่ 2 }, ... { รายการสมาชิกของตัวแปรชุดตัวที่ n } };
--------	--



ตัวอย่างการกำหนดค่าเริ่มต้นให้กับตัวแปรชุดโครงสร้าง

- การกำหนดค่าเริ่มต้นของตัวแปรชุดโครงสร้างจะต้องกำหนดค่าเริ่มต้นให้ตรงกับสมาชิกหรือฟิลด์ของโครงสร้างนั้นๆ

```
#include <stdio.h>
typedef struct{
    char name[50];
    int age;
    char grade;
}profile;
int main()
{
    profile student[30]={{ "Manee",18, 'C'},
                        {"Piti", 17, 'A'} ,
                        {"Chujai",18, 'B'}};

    return (0);
}
```



การเข้าถึงฟิลด์ข้อมูลของตัวแปรชุดโครงสร้าง

- การเข้าถึงฟิลด์ข้อมูลของตัวแปรชุดโครงสร้างต้องระบุดัชนีพร้อมชื่อฟิลด์

รูปแบบ	ชื่อตัวแปรชุดโครงสร้าง[ดัชนี] • ชื่อฟิลด์ข้อมูล
--------	---

- ดัชนี หมายถึง ตำแหน่งช่องข้อมูลชุดใดๆ ซึ่งภายในการเก็บข้อมูลแบบโครงสร้าง
 - ดัชนี เท่ากับ 0 หมายถึง ข้อมูลโครงสร้างชุดที่ 1 หรือ ข้อมูลโครงสร้างช่องแรก
- การเข้าถึงฟิลด์ข้อมูลของ struct ทำได้โดยใช้โอเปอเรเตอร์จุด (.) เพื่อไปกำหนดค่าหรือดึงค่าของฟิลด์ภายในตัวแปรโครงสร้างนั้นๆ เพื่อนำมาประมวลผล



การเข้าถึงฟิลด์ข้อมูลของตัวแปรชุดโครงสร้าง

- การเข้าถึงฟิลด์ข้อมูลของตัวแปรชุดโครงสร้างต้องระบุดัชนีพร้อมชื่อฟิลด์

```
typedef struct{
    char name[50];
    int age;
    char grade;
}profile;

profile student[30];
```

การเข้าถึงฟิลด์ภายในตัวแปรชุดโครงสร้าง student จะต้องเขียนคำสั่ง

```
strcpy(student[0].name, "sirinthorn");
student[0].age = 18;
student[0].grade = 'A';
strcpy(student[1].name, "cheyasak");
student[1].age = 17;
student[1].grade = 'B';
```



ตัวอย่างโปรแกรมการเก็บข้อมูลในตัวแปรชุดโครงสร้าง

```
#include <stdio.h>
typedef struct{
    char name[50];
    int age;
    char grade;
}profile;
int main()
{
    profile student[4];
    int i;
    printf("Please <Enter> for Input 3 records: \n");
    for(i=0;i<4;i++)
    {
        printf("-----\n");
        printf("RECORD # %d \n",i+1);
        printf("\tNAME : ");
        scanf("%s",&student[i].name);
        printf("\tAGE : ");
        scanf("%d",&student[i].age);
        printf("\tGRADE:");
        scanf(" %c",&student[i].grade);
    }
    return (0);
}
```



ตัวอย่างการรันโปรแกรม

```
C:\Dev-Cpp\bin\arr1.exe
Please <Enter> for Input 4 records:
-----
RECORD # 1
      NAME : Manee
      AGE  : 18
      GRADE: A
-----
RECORD # 2
      NAME : Piti
      AGE  : 17
      GRADE: B
-----
RECORD # 3
      NAME : Chujai
      AGE  : 18
      GRADE: C
-----
RECORD # 4
      NAME : Manah
      AGE  : 17
      GRADE: d
```



การเปรียบเทียบระหว่าง Parallel Arrays กับ Array of Structures

- ภาพการเก็บข้อมูลด้วย Parallel Array เมื่อเปรียบเทียบกับ Array of Structure

	Name	Age	Grade
[0]	“Manee”	18	‘A’
[1]	“Piti”	17	‘B’
[2]	“Chujai”	18	‘C’
[3]	“Manah”	17	‘D’

student

[0]	“Manee”	18	‘A’
[1]	“Piti”	17	‘B’
[2]	“Chujai”	18	‘C’
[3]	“Manah”	17	‘D’

การ swap ค่าของ Parallel Arrays

```
strcpy(bname,Name[0]);
```

```
strcpy(Name[0],Name[1]);
```

```
strcpy(Name[1],bname);
```

```
bage = Age[0];      bgrade = Grade[0];
```

```
Age[0] = Age[1];      Grade[0] = Grade[1];
```

```
Age[1] = bage;      Grade[1] = bgrade;
```

การ swap ค่าของ Array of Structure

```
bstudent = student[0];
```

```
student[0] = student[1];
```

```
student[1] = bstudent;
```



ตัวอย่างโปรแกรมการประยุกต์ใช้งานตัวแปรแบบโครงสร้าง

```
#include <stdio.h>
typedef struct{
    char name[50];
    int age;
    char grade;
}profile;
int main()
{
    profile student[3] = {"Manee", 17, 'C' } ,
                        {"Piti", 16, 'A' } ,
                        {"Chujai", 17, 'B'}};

    bstudent = student[0];
    student[0] = student[1];
    student[1] = bstudent;
    printf("student[0]: %s %d %c \n", student[0].name
                                                , student[0].age
                                                , student[0].grade);

    printf("student[1]: %s %d %c \n", student[1].name
                                                , student[1].age
                                                , student[1].grade);

    getch();
    return(0);
}
```



การใช้ตัวแปรชุดโครงสร้างกับฟังก์ชัน

- การสร้างฟังก์ชันโดยการรับค่าผ่านพารามิเตอร์เข้ามาเป็นโครงสร้าง
 - กรณีนิยามโครงสร้างด้วย typedef

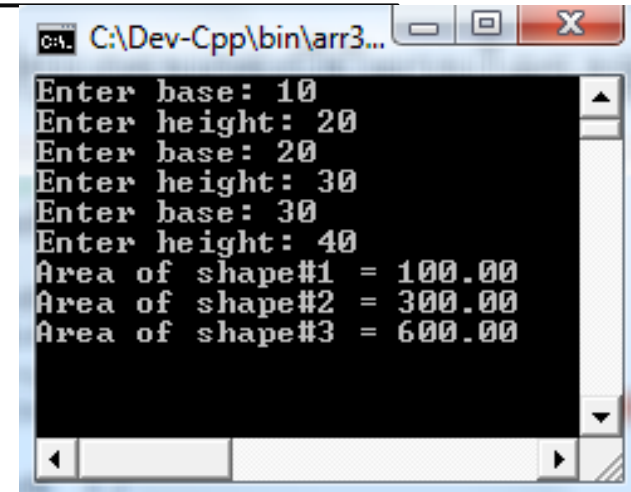
void ชื่อฟังก์ชัน(ชื่อโครงสร้าง ชื่อตัวแปรโครงสร้าง)

```
{  
  ...  
}
```



ตัวอย่างโปรแกรมกรณีนิยามโครงสร้างด้วย typedef

```
#include <stdio.h>
typedef struct shape
{
    int base,height;
    float area;
}triangle;
void GetData(triangle t[])
{
    int i;
    for(i=0; i<3; i++)
    {
        printf("Enter base: ");
        scanf("%d",&t[i].base);
        printf("Enter height: ");
        scanf("%d",&t[i].height);
    }
}
int main()
{
    triangle t[3];
    GetData(t);
    getch();
    return(0);
}
```



```
C:\Dev-Cpp\bin\arr3...
Enter base: 10
Enter height: 20
Enter base: 20
Enter height: 30
Enter base: 30
Enter height: 40
Area of shape#1 = 100.00
Area of shape#2 = 300.00
Area of shape#3 = 600.00
```



ตัวอย่างการใช้ตัวแปรชุดโครงสร้างกับฟังก์ชัน (ต่อ)

```
void Findarea(triangle t[]) ←  
{  
    int i;  
    for(i=0; i<3; i++)  
    {  
        t[i].area = 0.5 * t[i].base * t[i].height;  
    }  
}  
void Display(triangle t[]) ←  
{  
    int i;  
    for(i=0; i<3; i++)  
    {  
        printf("Area of shape#%d = %0.2f\n",i+1,t[i].area);  
    }  
}
```



การเก็บข้อมูลด้วยโครงสร้างซ้อนโครงสร้าง

- Structure written inside another structure is called as nesting of two structures.
- Nested Structures are allowed in C Programming Language.
- We can write one Structure inside another structure as member of another structure.



การประกาศโครงสร้างซ้อนโครงสร้าง

- การประกาศโครงสร้างซ้อนโครงสร้างมี 2 รูปแบบคือ
 - ประกาศแยกโครงสร้างระหว่างโครงสร้างหลักและโครงสร้างย่อย สำหรับวิธีนี้จะต้องประกาศโครงสร้างย่อยก่อนโครงสร้างหลักเสมอ
 - ประกาศโครงสร้างย่อยไว้ภายในโครงสร้างหลัก



การประกาศโครงสร้างซ้อนโครงสร้าง (แบบที่ 1)

- ประกาศแยกโครงสร้างระหว่างโครงสร้างหลักและโครงสร้างย่อย

```
struct date
{
    int date;
    int month;
    int year;
};

struct Employee
{
    char ename[20];
    int ssn;
    float salary;
    struct date doj;
}emp1;
```



การประกาศโครงสร้างซ้อนโครงสร้าง (แบบที่ 2)

- ประกาศโครงสร้างย่อยไว้ในโครงสร้างหลัก

```
struct Employee
{
    char ename[20];
    int ssn;
    float salary;
    struct date
        {
            int date;
            int month;
            int year;
        }doj;
}empl;
```



การเข้าถึงตัวแปรชนิดโครงสร้างซ้อนโครงสร้าง

- การอ้างถึงตัวแปรภายในโครงสร้างซ้อนโครงสร้างจะใช้รูปแบบดังนี้

ตัวแปรโครงสร้างหลัก . ตัวแปรโครงสร้างย่อย . ตัวแปรสมาชิก(ฟิลด์ย่อย)



การเข้าถึงตัวแปรโครงสร้างซ้อนโครงสร้าง (แบบที่ 1)

- ประกาศแยกโครงสร้างระหว่างโครงสร้างหลักและโครงสร้างย่อย

```
struct date
{
    int date;
    int month;
    int year;
};

struct Employee
{
    char ename[20];
    int ssn;
    float salary;
    struct date doj;
}emp1;
```

- การเข้าถึงฟิลด์ month ด้วยคำสั่ง emp1.doj.month
- การเข้าถึงฟิลด์ day ด้วยคำสั่ง emp1.doj.day
- การเข้าถึงฟิลด์ year ด้วยคำสั่ง emp1.doj.year



การเข้าถึงตัวแปรโครงสร้างซ้อนโครงสร้าง (แบบที่ 2)

- ประกาศแยกโครงสร้างระหว่างโครงสร้างหลักและโครงสร้างย่อย

```
struct Employee
{
    char ename[20];
    int ssn;
    float salary;
    struct date
    {
        int date;
        int month;
        int year;
    }doj;
}emp1;
```

- การเข้าถึงฟิลด์ month ด้วยคำสั่ง emp1.doj.month
- การเข้าถึงฟิลด์ day ด้วยคำสั่ง emp1.doj.day
- การเข้าถึงฟิลด์ year ด้วยคำสั่ง emp1.doj.year



ตัวอย่างโปรแกรมการประยุกต์ใช้โครงสร้างซ้อนโครงสร้าง

```
#include <stdio.h>

struct Employee
{
    char ename[20];
    int ssn;
    float salary;
    struct date
    {
        int date;
        int month;
        int year;
    }doj;
}emp = {"Pritesh",1000,1000.50,{22,6,1990}};

int main(int argc, char *argv[])
{
    printf("\nEmployee Name      : %s",emp.ename);
    printf("\nEmployee SSN          : %d",emp.ssn);
    printf("\nEmployee Salary       : %f",emp.salary);
    printf("\nEmployee DOJ          : %d/%d/%d", \
        emp.doj.date,emp.doj.month,emp.doj.year);

    return 0;
}
```

Output :

```
Employee Name      : Pritesh
Employee SSN       : 1000
Employee Salary    : 1000.500000
Employee DOJ       : 22/6/1990
```



สรุปท้ายบทเรียน

- ตัวแปรแบบชุดโครงสร้าง (Array of Structures)
 - ตัวแปรแบบชุดโครงสร้างสามารถรองรับการเก็บข้อมูลได้หลายชุดในแต่ละชุด ข้อมูลสามารถเก็บข้อมูลได้หลากหลายชนิดข้อมูล โดยผู้ใช้สามารถนิยามโครงสร้างได้เอง
 - การเข้าถึงฟิลด์ข้อมูลของตัวแปรชุดโครงสร้างต้องระบุดัชนีพร้อมชื่อฟิลด์
 - ดัชนี หมายถึง ตำแหน่งข้อมูลชุดใดๆ ซึ่งภายในการเก็บข้อมูลแบบโครงสร้าง การระบุดัชนีทำได้โดยใช้สัญลักษณ์ []
 - ดัชนี เท่ากับ 0 หรือ [0] หมายถึง ข้อมูลโครงสร้างชุดที่ 1 หรือ ข้อมูลโครงสร้างช่องแรก
 - การเข้าถึงฟิลด์ข้อมูลของโครงสร้างทำได้โดยใช้โอเปอเรเตอร์จุด (.) เพื่อไปกำหนดค่าหรือดึงค่าของฟิลด์ภายในตัวแปรโครงสร้างนั้นๆ เพื่อนำมาประมวลผล



สรุปท้ายบทเรียน (ต่อ)

- ตัวแปรแบบโครงสร้างซ้อนโครงสร้าง (Nested Structures)
 - ประกอบด้วยโครงสร้างหลักและโครงสร้างย่อยซ้อนภายในทั้งนี้เพื่อจัดกลุ่มชนิดข้อมูลพื้นฐานเป็นกลุ่มโครงสร้างย่อยเพื่อสะดวกสำหรับการเข้าถึงและจัดการได้โดยการอ้างอิงถึงกลุ่มโครงสร้างย่อย
 - การประกาศโครงสร้างซ้อนโครงสร้างมี 2 ลักษณะ คือ
 - ประกาศแยกระหว่างโครงสร้างหลักและโครงสร้างย่อย
 - ประกาศโครงสร้างย่อยไว้ภายในโครงสร้างหลัก
 - การเข้าถึงฟิลด์หรือสมาชิกข้อมูลภายในโครงสร้างย่อย คือ
ตัวแปรโครงสร้างหลัก . ตัวแปรโครงสร้างย่อย . ตัวแปรสมาชิก(ฟิลด์ย่อย)



เอกสารอ้างอิง

- ตำราอ้างอิง

สมจิตต์ ลิขิตถาวร (2004). โครงสร้างข้อมูล&การวิเคราะห์อัลกอริทึมด้วยซี. บริษัทซี เอ็ดดูเคชั่นจำกัด กรุงเทพฯ.

Stabdish, Thomas A.(1994). Data structures Algorithms and Software. MA: Addison-Wesley.

Lipschutz, S.(1986). Schaum s Outline of Theory and Problems of Data structures. Sigapore:McGraw-Hill.

- เว็บไซต์อ้างอิง

- <http://www.cp.eng.chula.ac.th/~somchai/2110211/2548/>

- http://dusithost.dusit.ac.th/~phorramatpanyaprat_ton/course/C4122202/c4122202.html

- <http://fresh2refresh.com/c/c-nested-structure/>

- <http://www.c4learn.com/c-programming/c-nested-structure/>

