



CS311: Computer Programming II

Week2: Structure in C

Website: http://bit.ly/sirinthorn_c

A.Sirinthorn Cheyasak

sirinthorn.c@bu.ac.th

School of Science and Technology, Bangkok University

วัตถุประสงค์การเรียนรู้

- เพื่อให้รู้จักโครงสร้างภาษาซี เข้าใจรูปแบบการจัดเก็บข้อมูลในโครงสร้าง
- เพื่อศึกษาและสามารถสร้างโครงสร้างพร้อมสามารถนำโครงสร้างไปประยุกต์ใช้สำหรับเก็บข้อมูลเพื่อนำไปประมวลผลได้
- เพื่อให้สามารถส่งผ่านค่าของโครงสร้างไปยังฟังก์ชันด้วยพารามิเตอร์ และคืนค่าโครงสร้างกลับผ่านฟังก์ชันได้



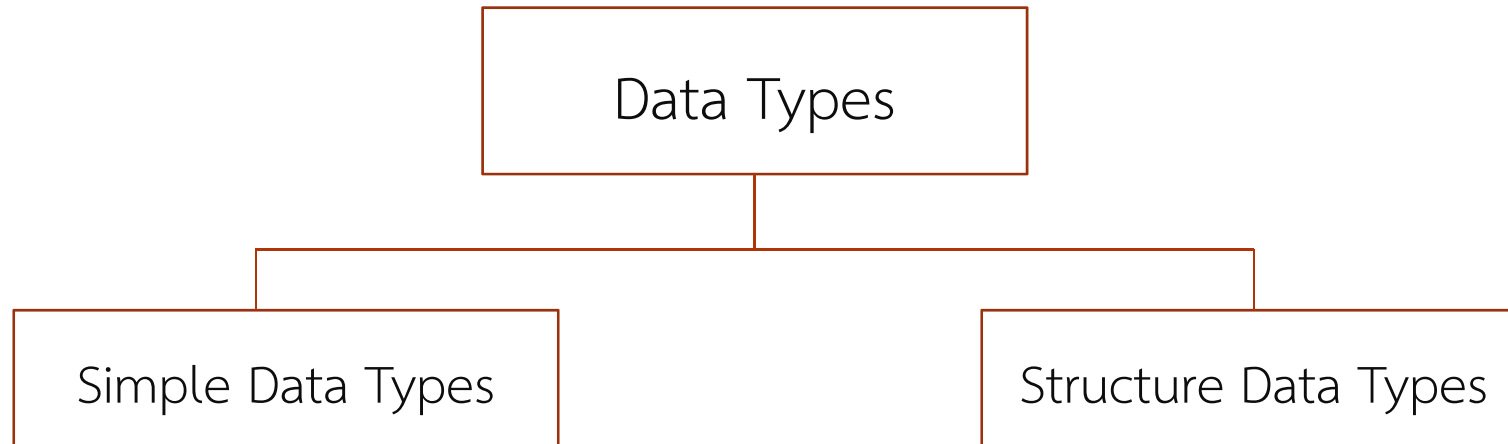
รายละเอียดเนื้อหา (Outline)

- โครงสร้าง (Structure) คืออะไร
- การนิยามโครงสร้างและการประกาศตัวแปรโครงสร้าง
- การเข้าถึงสมาชิกในโครงสร้าง
- การคัดลอกค่าตัวแปรโครงสร้าง
- การให้ค่าเริ่มต้นกับตัวแปรโครงสร้าง
- การกำหนดชนิดตัวแปรใหม่
- การส่งผ่านโครงสร้างไปยังฟังก์ชัน



การจัดกลุ่มประเภทของข้อมูลในภาษาซี

- ภาษาซีแบ่งประเภทของชนิดข้อมูลออกเป็น 2 ประเภทหลัก



- Simple Data Types หมายถึงชนิดข้อมูลพื้นฐาน เช่น int, char, float, etc.
- Structure Data Types หมายถึงชนิดข้อมูลแบบโครงสร้าง โดยการประกาศโครงสร้างใหม่ด้วยการนำข้อมูลพื้นฐานมารวมกัน จึงสามารถเก็บข้อมูลได้หลากหลายที่อาจจะเป็นชนิดเดียวกัน หรือต่างชนิดกัน หรือภายในโครงสร้างเอง อาจจะสามารถเป็นอาร์เรย์หรือแม้แต่มีสมาชิกเป็นโครงสร้างได้อีกด้วย



ทำไมจึงต้องเก็บข้อมูลแบบโครงสร้าง (Structure)

- ข้อมูลปัจจุบันมีความหลากหลายและมีจำนวนมากขึ้น เช่น ข้อมูลนักเรียน ข้อมูลลูกค้า ข้อมูลสินค้า
 - ข้อมูลนักเรียน ได้แก่ รหัสนักเรียน ชื่อ ชั้นปี เกรดเฉลี่ย เป็นต้น
 - ข้อมูลลูกค้า ได้แก่ รหัสลูกค้า ชื่อ ที่อยู่ ยอดซื้อ เป็นต้น
 - ข้อมูลสินค้า ได้แก่ รหัสสินค้า ชื่อสินค้า ประเภทสินค้า ราคา เป็นต้น
- เพื่อรองรับการเก็บข้อมูลที่ซับซ้อนมากขึ้น ภาษาซีจึงมีชนิดข้อมูลประเภทโครงสร้าง เพื่อให้ผู้เขียนโปรแกรมสามารถออกแบบโครงสร้างข้อมูลได้เองผ่านการประกาศโครงสร้างก่อนจะนำโครงสร้างข้อมูลนี้ไปประกาศตัวแปรเพื่อนำไปประมวลผลต่อไป



การประกาศโครงสร้างเพื่อเก็บข้อมูลด้วย struct

- การประกาศข้อมูลโครงสร้างข้อมูล มีรูปแบบดังนี้

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

- การประกาศโครงสร้างข้อมูล เป็นเพียงการประกาศหรือกำหนดชื่อของโครงสร้างข้อมูล เพื่อที่จะนำไปใช้ประกาศตัวแปรสำหรับนำไปประมวลผล



การประกาศโครงสร้างเพื่อเก็บข้อมูล (ต่อ)

- ตัวอย่างการประกาศโครงสร้าง
- โครงสร้าง address สำหรับเก็บข้อมูลที่อยู่ซึ่งประกอบด้วยฟิลด์สำหรับเก็บข้อมูลจำนวน 4 ฟิลด์
- โครงสร้าง student สำหรับเก็บข้อมูลนักเรียนซึ่งประกอบด้วย 3 ฟิลด์

ชื่อโครงสร้างข้อมูล

```
struct address{  
    char name[30];  
    char detail[50];  
    int age;  
    char telephone[10];  
};
```

ชื่อโครงสร้างข้อมูล

```
struct student {  
    int id;  
    char name[30];  
    char surname[50];  
};
```



ตัวอย่างการประกาศโครงสร้างและตัวแปรโครงสร้าง

```
#include <stdio.h>
struct student {
    int id;
    char name[30];
    char surname[50];
};

int main()
{
    struct student st1, st2;

    return (0);
}
```

- ตัวอย่างการประกาศโครงสร้าง student ประกอบด้วยฟิลด์ข้อมูล 3 ฟิลด์ ได้แก่ id เก็บข้อมูลตัวเลขจำนวนเต็ม, name เก็บข้อมูลชื่อด้วยสตริง และ surname เก็บข้อมูลนามสกุลด้วยสตริง พร้อมประกาศตัวแปรโครงสร้างภายในฟังก์ชันหลัก (main)



การประกาศโครงสร้างเพื่อเก็บข้อมูลด้วย struct

- การประกาศโครงสร้างข้อมูลพร้อมประกาศตัวแปร มีรูปแบบดังนี้

รูปแบบที่ 2	<pre>struct ชื่อโครงสร้างข้อมูล { ชนิดข้อมูล ชื่อฟิลด์ข้อมูลที่ 1; ชนิดข้อมูล ชื่อฟิลด์ข้อมูลที่ 2; ... ชนิดข้อมูล ชื่อฟิลด์ข้อมูลที่ n; }variable-name,[variable-name];</pre>
-------------	--

- การประกาศโครงสร้างข้อมูล เป็นเพียงการประกาศหรือกำหนดชื่อของโครงสร้างข้อมูล เพื่อที่จะนำไปใช้กำหนดให้เป็นชนิดของตัวแปร



ตัวอย่างการประกาศโครงสร้างและตัวแปรโครงสร้าง

```
#include <stdio.h>
struct student {
    int id;
    char name[30];
    char surname[50];
}st1, st2;

int main()
{

    return (0) ;
}
```

- ตัวอย่างการประกาศโครงสร้าง student ประกอบด้วยฟิลด์ข้อมูล 3 ฟิลด์ ได้แก่ id เก็บข้อมูลตัวเลขจำนวนเต็ม, name เก็บข้อมูลชื่อด้วยสตริง และ surname เก็บข้อมูลนามสกุลด้วยสตริง พร้อมประกาศตัวแปร st1 และ st2 ต่อจากการประกาศโครงสร้าง แต่จะทำให้ตัวแปรนี้เป็นตัวแปร Global



การประกาศโครงสร้างเพื่อเก็บข้อมูลด้วย typedef

- การประกาศโครงสร้างข้อมูลด้วย typedef มีรูปแบบดังนี้

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

- การประกาศโครงสร้างข้อมูล เป็นเพียงการประกาศหรือกำหนดชื่อของโครงสร้างข้อมูล เพื่อที่จะนำไปใช้กำหนดให้เป็นชนิดของตัวแปร



ตัวอย่างการประกาศโครงสร้างและตัวแปรโครงสร้าง

```
#include <stdio.h>
typedef struct{
    int id;
    char name[30];
    char surname[50];
} student;

int main()
{
    student st1, st2;
    return (0);
}
```

- ตัวอย่างการประกาศโครงสร้างโดยใช้ typedef ชื่อโครงสร้าง student ประกอบด้วยฟิลด์ข้อมูล 3 ฟิลด์ ได้แก่ id เก็บข้อมูลตัวเลขจำนวนเต็ม, name เก็บข้อมูลชื่อด้วยสตริง และ surname เก็บข้อมูลนามสกุลด้วยสตริง พร้อมประกาศตัวแปรโครงสร้างด้วยชื่อโครงสร้าง student เท่านั้นโดยไม่ต้องใช้คีย์เวิร์ด struct อีก



เปรียบเทียบการประกาศโครงสร้าง

- ตัวอย่างการเปรียบเทียบการประกาศโครงสร้างระหว่าง struct และ typedef
- การประกาศโครงสร้าง profile สำหรับเก็บข้อมูลพนักงานซึ่งประกอบด้วย 4 필ด์

ชื่อโครงสร้างข้อมูล

```
struct profile{  
    char fname[30];  
    char lname[50];  
    int age,status;  
};  
struct profile emp1,emp2;
```

```
typedef struct{  
    char fname[30];  
    char lname[50];  
    int age,status;  
}profile;
```

ชื่อโครงสร้างข้อมูล

```
profile emp1,emp2;
```



การกำหนดค่าเริ่มต้นให้กับตัวแปรโครงสร้าง

- การกำหนดค่าเริ่มต้นของตัวแปรโครงสร้างจะต้องกำหนดค่าเริ่มต้นให้ตรงกับสมาชิกหรือฟิลด์ของโครงสร้างนั้นๆ

ชื่อโครงสร้างข้อมูล

```
struct profile{  
    char fname[30];  
    char lname[50];  
    int age,status;  
};
```

โครงสร้าง profile ประกอบด้วย 4 ฟิลด์

- fname สำหรับเก็บชื่อ 30 ตัวอักษร
- lname สำหรับเก็บนามสกุล 50 ตัวอักษร
- age สำหรับเก็บอายุ
- status สำหรับเก็บสถานะพนักงาน

```
struct profile emp1={"Sirinthorn","Cheyasak",35,1};
```



การเข้าถึงฟิลด์ข้อมูลของตัวแปรโครงสร้าง

- การเข้าถึงฟิลด์ข้อมูล

รูปแบบ	ชื่อตัวแปรโครงสร้าง.ชื่อฟิลด์ข้อมูล
--------	-------------------------------------

- การเข้าถึงฟิลด์ข้อมูลของ struct ทำได้โดยใช้โอเปอเรเตอร์จุด (.) เพื่อไปกำหนดค่าหรือดึงค่าของฟิลด์ภายในตัวแปรโครงสร้างนั้นๆ เพื่อนำมาประมวลผล

```
typedef struct{
    char fname[30];
    char lname[50];
    int age,status;
}profile;

profile emp1,emp2;
```

การเข้าถึงฟิลด์ fname ภายในตัวแปร emp1
จะต้องเขียนคำสั่ง

```
strcpy(emp1.fname,"sirinthorn");
strcpy(emp1.lname,"cheyasak");
emp1.age = 35;
emp1.status = 2;
```



ตัวอย่างการกำหนดฟิลด์ที่มีชนิดข้อมูลเป็นอาร์เรย์

รูปแบบ

ชื่อตัวแปรโครงสร้าง • ชื่อฟิลด์ข้อมูลอาร์เรย์ [ดัชนี]

```
struct employee{
    char name[25];
    int age;
    int pay;
};
int main()
{
    struct employee emp1, emp2;
    emp2.name[0] = 'S';
    emp2.name[1] = 'i';
    emp2.name[2] = 'r';
    emp2.name[3] = 'i';
    emp2.name[4] = 'n';
    emp2.name[5] = '\0';
    printf("Name: %s\n", emp2.name);
    getch();
    return(0);
}
```



ตัวอย่างการกำหนดค่าสำหรับฟิลด์ที่มีชนิดข้อมูลเป็นอาร์เรย์(ต่อ)

รูปแบบ

ชื่อตัวแปรโครงสร้าง • ชื่อฟิลด์ข้อมูลอาร์เรย์ [ดัชนี]

```
struct score {
    float midterm;
    float final;
    int lab[3];
} student;
int main()
{
    student.midterm = 23.5;
    student.final = 30.0;
    student.lab[0] = 78;
    student.lab[1] = 82;
    student.lab[2] = 90;
    return (0);
}
```



การคัดลอกค่าของตัวแปรโครงสร้าง

- การคัดลอก (Copy) ค่าของตัวแปรโครงสร้างชนิดเดียวกัน สามารถทำได้ โดยใช้เครื่องหมาย “=” (เท่ากับ)

```
struct date{  
    int day,month,year;  
} d1,d2;
```

```
d2.day = d1.day;  
d2.month = d1.month;  
d2.year = d1.year;
```

- การคัดลอกค่าของทุกฟิลด์ผ่านตัวแปรโครงสร้างได้เลย กรณีมีโครงสร้างเดียวกัน

```
d2 = d1;
```



ตัวอย่างโปรแกรมการประยุกต์ใช้งานตัวแปรแบบโครงสร้าง

```
struct score {
    float midterm;
    float final;
    int lab[3];
};
int main()
{
    struct score st1,st2;
    FILE *inf;
    int i;
    inf = fopen("score.txt", "r");
    fscanf(inf, "%f%f", &st1.midterm, &st1.final);
    for(i=0; i<3; i++)
    {
        fscanf(inf, "%d", &st1.lab[i]);
    }
    return (0);
}
```



การใช้ตัวแปรชนิดโครงสร้างกับฟังก์ชัน

- การใช้งานตัวแปรโครงสร้างกับฟังก์ชันสามารถทำได้หลายลักษณะเช่นเดียวกับการใช้งานตัวแปรพื้นฐาน
 - การรับข้อมูลโครงสร้างเข้าสู่ฟังก์ชันผ่านพารามิเตอร์ (Parameter passing)
 - การส่งข้อมูลกลับผ่านฟังก์ชันด้วยตัวแปรโครงสร้าง (Return structure with function return-type)



การรับข้อมูลโครงสร้างเข้าสู่ฟังก์ชันผ่านพารามิเตอร์

- การสร้างฟังก์ชันโดยการรับค่าผ่านพารามิเตอร์เข้ามาเป็นโครงสร้าง
 - กรณีนียามโครงสร้างด้วย struct

void ชื่อฟังก์ชัน(struct ชื่อโครงสร้าง ชื่อตัวแปรโครงสร้าง)

```
{  
  ...  
}
```

- กรณีนียามโครงสร้างด้วย typedef

void ชื่อฟังก์ชัน(ชื่อโครงสร้าง ชื่อตัวแปรโครงสร้าง)

```
{  
  ...  
}
```



ตัวอย่างโปรแกรมกรณีนิยามโครงสร้างด้วย struct

```
#include <stdio.h>
struct triangleshape ←
{
    int base,height;
};
void findareal(struct triangleshape t)
{
    float area;
    area = 0.5 * t.base * t.height;
    printf("Area of a triangle = %0.2f\n",area);
    printf("Calculated by findareal function\n");
}
int main()
{
    struct triangleshape t1;
    printf("Enter base: ");
    scanf("%d",&t1.base);
    printf("Enter height: ");
    scanf("%d",&t1.height);
    findareal(t1);
    getch();
    return(0);
}
```



ตัวอย่างโปรแกรมกรณีนิยามโครงสร้างด้วย typedef

```
#include <stdio.h>
typedef struct shape
{
    int base,height;
}triangle;
void findarea2(triangle t)
{
    float area;
    area = 0.5 * t.base * t.height;
    printf("Area of a triangle = %0.2f\n",area);
    printf("Calculated by findarea2 function\n");
}
int main()
{
    triangle t2;
    printf("Enter base: ");
    scanf("%d",&t2.base);
    printf("Enter height: ");
    scanf("%d",&t2.height);
    findarea2(t2);
    getch();
    return(0);
}
```



การส่งข้อมูลกลับผ่านฟังก์ชันด้วยตัวแปรโครงสร้าง

- การส่งข้อมูลกลับผ่านฟังก์ชันด้วยตัวแปรโครงสร้าง

- กรณีนิยามโครงสร้างด้วย struct

struct ชื่อโครงสร้าง ชื่อฟังก์ชัน(struct ชื่อโครงสร้าง ชื่อตัวแปรโครงสร้าง)

```
{  
  ...  
}
```

- กรณีนิยามโครงสร้างด้วย typedef

ชื่อโครงสร้าง ชื่อฟังก์ชัน(ชื่อโครงสร้าง ชื่อตัวแปรโครงสร้าง)

```
{  
  ...  
}
```



ตัวอย่างโปรแกรมกรณีนิยามโครงสร้างด้วย struct

```
#include <stdio.h>
struct triangleshape
{
    int base,height;
    float area;
};
struct triangleshape findareal(struct triangleshape t)
{
    t.area = 0.5 * t.base * t.height;
    return(t);
}
int main()
{
    struct triangleshape t1;
    printf("Enter base: ");
    scanf("%d",&t1.base);
    printf("Enter height: ");
    scanf("%d",&t1.height);
    t1 = findareal(t1);
    printf("Area of a triangle = %0.2f\n",t1.area);
    printf("Calculated by findareal function\n");
    getch();
    return(0);
}
```



ตัวอย่างโปรแกรมกรณีนิยามโครงสร้างด้วย typedef

```
#include <stdio.h>
typedef struct shape
{
    int base,height;
    float area;
}triangle;
triangle findarea2(triangle t)
{
    t.area = 0.5 * t.base * t.height;
    return(t);
}
int main()
{
    triangle t2;
    printf("Enter base: ");
    scanf("%d",&t2.base);
    printf("Enter height: ");
    scanf("%d",&t2.height);
    t2 = findarea2(t2);
    printf("Area of a triangle = %0.2f\n",t2.area);
    printf("Calculated by findarea2 function\n");
    getch();
    return(0);
}
```



สรุปท้ายบทเรียน

- ตัวแปรแบบโครงสร้างสามารถรองรับชนิดข้อมูลได้หลากหลาย โดยผู้ใช้สามารถนิยามโครงสร้างได้เอง ดังนี้
 - การนิยามโครงสร้างด้วย struct
 - การนิยามโครงสร้างด้วย typedef
- การนิยามหรือประกาศโครงสร้างใหม่เกิดประกอบด้วยการนำข้อมูลพื้นฐานมารวมกัน ทำให้สามารถเก็บข้อมูลได้หลากหลายที่อาจจะเป็นชนิดเดียวกัน หรือต่างชนิดกัน หรือภายในโครงสร้างเองอาจจะสมาชิกเป็นอาร์เรย์หรือแม้แต่มีสมาชิกเป็นโครงสร้างได้อีก
- การเข้าถึงฟิลด์หรือสมาชิกภายในโครงสร้างจะใช้สัญลักษณ์คือจุด (.)



สรุปท้ายบทเรียน (ต่อ)

- การใช้งานตัวแปรโครงสร้างกับฟังก์ชันสามารถทำได้หลายลักษณะเช่นเดียวกับการใช้งานตัวแปรพื้นฐาน
 - การรับข้อมูลโครงสร้างเข้าสู่ฟังก์ชันผ่านพารามิเตอร์ (Parameter passing)
 - การส่งข้อมูลกลับผ่านฟังก์ชันด้วยตัวแปรโครงสร้าง (Return structure with function return-type)



เอกสารอ้างอิง

- ตำราอ้างอิง

สมจิตต์ ลิขิตถาวร (2004). โครงสร้างข้อมูล&การวิเคราะห์อัลกอริทึมด้วยซี.
บริษัทซีเอ็ดยูเคชั่นจำกัด กรุงเทพฯ.

Stabdish, Thomas A.(1994). Data structures Algorithms and
Software. MA: Addision-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/>

