

STRUCTURE

ตัวแปรโครงสร้าง

1

รูปแบบการเก็บข้อมูล

ตัวแปรพื้นฐาน

a

5

ตัวแปรแถวลำดับ

arr

arr[0]

arr[1]

arr[2]

arr[3]

5

15

12

1

2

รูปแบบการเก็บข้อมูล

ตัวแปรแถวลำดับ

arr

arr[0]

arr[1]

arr[2]

arr[3]

5	15	12	1
---	----	----	---

ตัวแปรโครงสร้าง

str

id

name

age

sex

15	Charuk	35	m
----	--------	----	---

3

รูปแบบการเก็บข้อมูล

แถวลำดับ:array

arr[0] arr[1] arr[2] arr[3]

--	--	--	--

โครงสร้าง:structure

id

name

age

sex

--	--	--	--

- เก็บข้อมูลได้หลายค่า (array/structure)
- ใช้หน่วยความจำเรียงต่อกัน (array/structure)
- เก็บข้อมูลได้หลากหลายชนิด (array/structure)
- เก็บข้อมูลชนิดเดียวกันหลายตัว (array/structure)
- เข้าถึงสมาชิกในชุดข้อมูลโดยใช้เลขดัชนี (array/structure)
- เข้าถึงสมาชิกในชุดข้อมูลโดยใช้ชื่อ (array/structure)

4

รูปแบบการเก็บข้อมูล : structure

การเก็บข้อมูลเราอาจจำเป็นต้องใช้ข้อมูลที่หลากหลายรูปแบบแต่เป็นกลุ่มข้อมูลที่มีส่วนร่วมกัน เช่น รายละเอียดของนักเรียนอาจจะมีข้อมูลรหัส ชื่อ สกุล อายุ เกรดเฉลี่ย ของนักเรียน ซึ่งเราจะต้องสร้างตัวแปรขึ้นมาใช้ในการเก็บข้อมูลของนักเรียน 5 ตัวแปรด้วยกัน ทั้งที่จริงแล้วข้อมูลทั้งหมดมีส่วนเกี่ยวข้องคือเป็นของนักเรียนคนเดียวกัน หากมีจำนวนนักเรียนมากๆ การสร้างตัวแปรและใช้งานตัวแปร อาจจะทำให้เกิดความสับสนขึ้นได้ ดังนั้นเราจึงควรนำตัวแปรโครงสร้างมาช่วยในการจัดการกับข้อมูลเหล่านี้

5

- อะเรย์ (array)

- เก็บข้อมูลจำนวนมากๆได้
- เข้าถึงข้อมูลแต่ละตัวได้โดยง่าย
- ข้อจำกัด : ข้อมูลเหล่านั้นต้องเป็นชนิดเดียวกันเท่านั้น

เช่น `int point[5]`

- ข้อมูลแบบโครงสร้าง

- ข้อมูลแต่ละตัวในกลุ่มสามารถมีชนิดต่างกันได้ เช่น
ข้อมูลผลการเรียนของนักเรียนแต่ละคน
- เทียบเท่ากับ Record ในภาษาอื่น

```
struct student {  
    char stdName[20];  
    int age;  
    float grade;  
}
```

6

ตัวแปรโครงสร้าง (Structure)

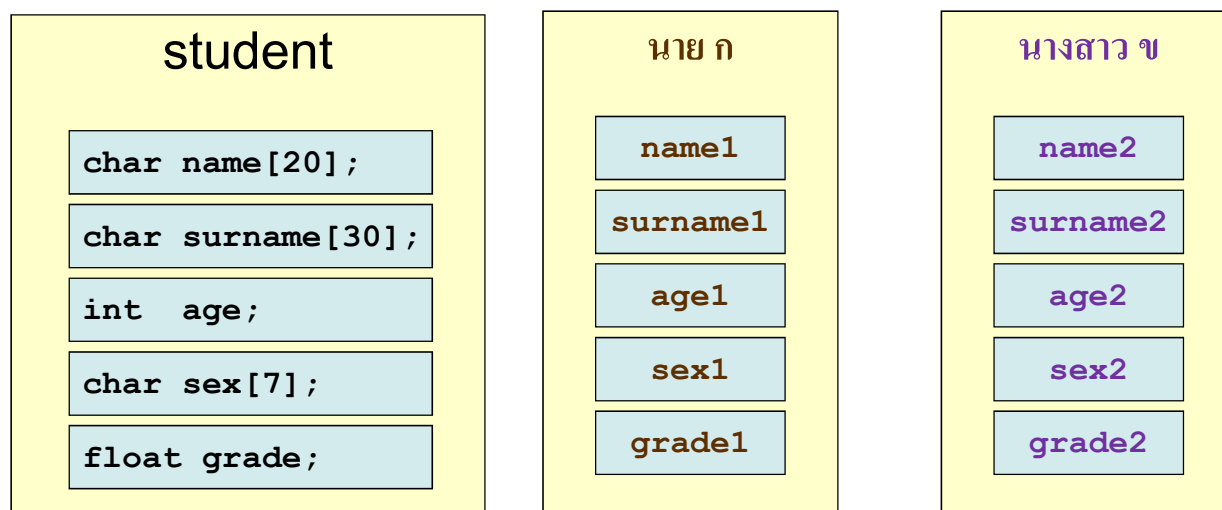
- ตัวแปรโครงสร้าง (Structure) เป็นการกำหนดชนิดข้อมูลใหม่ขึ้นมาใช้งานโดยชนิดข้อมูลนี้เกิดจากการรวมกันของชนิดข้อมูลพื้นฐานที่อาจจะเป็นชนิดเดียวกันหรือต่างชนิดกันก็ได้

ตัวอย่าง

ตัวแปรโครงสร้างนักเรียนให้ชื่อว่า student ประกอบด้วย

ชื่อ	เป็นชนิด	char
นามสกุล	เป็นชนิด	char
อายุ	เป็นชนิด	int
เพศ	เป็นชนิด	char
เกรดเฉลี่ย	เป็นชนิด	float

7



8

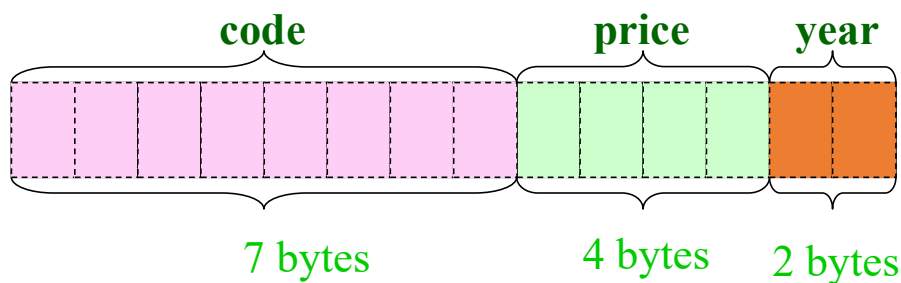
การกำหนดตัวแปรโครงสร้าง

- ก่อนการกำหนดตัวแปรชนิดโครงสร้างเพื่อมาใช้งาน เราจะต้องกำหนดตัวแปรโครงสร้างสำหรับตัวแปร
- โดยการกำหนดตัวแปรโครงสร้างในภาษา C มีข้อกำหนดดังนี้

```
struct ชื่อตัวแปรโครงสร้าง {  
    ชนิดข้อมูล ชื่อข้อมูลที่หนึ่ง;  
    ชนิดข้อมูล ชื่อข้อมูลที่สอง;  
    ...  
    ชนิดข้อมูล ชื่อข้อมูลที่ n;  
};
```

9

```
struct book {  
    char code[7];  
    float price;  
    int year;  
};
```



10

ตัวอย่างการกำหนดตัวแปรโครงสร้าง

ตัวแปรโครงสร้างสำหรับเก็บข้อมูลนักเรียน ซึ่ง ประกอบด้วย
ชื่อ เป็นข้อความไม่เกิน 20 ตัวอักษร
นามสกุล เป็นข้อความไม่เกิน 30 ตัวอักษร
อายุ เป็นจำนวนเต็ม
เพศ เป็นข้อความยาวไม่เกิน 7 ตัวอักษร
ระดับคะแนนเฉลี่ย เป็นจำนวนจริง

ตัวแปรโครงสร้างสำหรับเก็บข้อมูล วันที่ ซึ่ง ประกอบด้วย
วันที่ เดือน และ ปี เป็นจำนวนเต็ม

ตัวแปรโครงสร้างสำหรับเก็บข้อมูล วันที่ ซึ่ง ประกอบด้วย
วันที่ เป็นข้อความ 2 ตัวอักษร
เดือน เป็นข้อความ 3 ตัวอักษร
ปี เป็นข้อความ 5 ตัวอักษร

11

ตัวอย่างการกำหนดตัวแปรโครงสร้าง

```
struct student {  
    char    name[20];  
    char    surname[30];  
    int     age;  
    char    sex[7];  
    float   grade;  
};
```

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

```
struct sdate {  
    char    day[2];  
    char    month[3];  
    char    year[5];  
};
```

12

การกำหนดตัวแปรชนิดโครงสร้าง

- เมื่อกำหนดตัวแปรโครงสร้างเรียบร้อยแล้ว เราสามารถกำหนดให้ตัวแปรใดๆ ของเรามีโครงสร้างตามที่เรากำหนดไว้แล้วได้โดยใช้รูปแบบต่อไปนี้

```
struct ชื่อตัวแปรโครงสร้าง ชื่อตัวแปร;
```

ตัวอย่าง

```
struct student student1;  
struct student Manee, Peeti;  
struct date birthday;
```

13

การกำหนดตัวแปรและโครงสร้าง

- การประกาศตัวแปรอาจทำพร้อมการกำหนดตัวแปรโครงสร้างเลยก็ได้ โดยใช้รูปแบบดังนี้

```
struct ชื่อตัวแปรโครงสร้าง {  
    ชนิดข้อมูล ชื่อสมาชิกที่หนึ่ง;  
    ชนิดข้อมูล ชื่อสมาชิกที่สอง;  
    ...  
    ชนิดข้อมูล ชื่อสมาชิกที่ n;  
} ชื่อของตัวแปร;
```

14

ตัวอย่างการกำหนดตัวแปรและตัวแปรโครงสร้าง

```
struct student {  
    char    name[20] ;  
    char    surname[30] ;  
    int     age ;  
    char    sex[7] ;  
    float   grade ;  
} student1 ;
```

```
struct date {  
    int day,month,year ;  
} vacation, birthday ;
```

15

การกำหนดตัวแปรและตัวแปรโครงสร้าง

กำหนดตัวแปรโครงสร้าง นักเรียน (student) ซึ่งประกอบด้วย
รหัสนักเรียน (id) เป็นจำนวนเต็มไม่เกิน 5 หลัก
ชื่อ (name) เป็นข้อความ ไม่เกิน 20 ตัวอักษร
คะแนนเก็บ (score) เป็นจำนวนเต็ม ไม่เกิน 60 คะแนน
คะแนนสอบกลางภาค (mid) เป็นจำนวนเต็ม ไม่เกิน 20 คะแนน
คะแนนสอบปลายภาค (final) เป็นจำนวนเต็ม ไม่เกิน 20 คะแนน
คะแนนรวม (sum) เป็นจำนวนเต็ม ไม่เกิน 100 คะแนน
แล้วสร้างตัวแปรชื่อ std1 และ std2 เป็นตัวแปรประเภทนี้

16

การกำหนดตัวแปรและตัวแปรโครงสร้าง

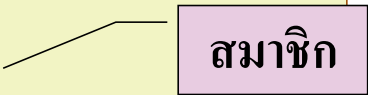
```
struct student{  
    int id;  
    char name[20];  
    int score;  
    int mid;  
    int final;  
    int sum;  
}std1,std2;
```

17

สมาชิกในตัวแปรชนิดโครงสร้าง

- พิจารณา ส่วนประกอบต่างๆ ในโครงสร้าง เราเรียกส่วนประกอบเหล่านี้ว่า “สมาชิก”

```
struct student {  
    char name[20];  
    char surname[30];  
    int age;  
    char sex[7];  
    float grade;  
} student1;
```



18

การอ้างถึงสมาชิกในตัวแปรชนิดโครงสร้าง

- การอ้างถึงสมาชิกในตัวแปรชนิดโครงสร้าง ทำได้โดยบอกชื่อตัวแปรโครงสร้างตามด้วยจุดและต่อด้วยชื่อสมาชิกของโครงสร้างนั้นๆ
- มีรูปแบบดังนี้

ชื่อตัวแปรโครงสร้าง.ชื่อสมาชิก;

ตัวอย่าง

<code>student1.name</code>	อ้างถึงข้อมูล <u>ชื่อ</u> ของตัวแปร <code>student1</code>
<code>student1.surname</code>	อ้างถึงข้อมูล <u>สกุล</u> ของตัวแปร <code>student1</code>
<code>birthday.day</code>	อ้างถึงข้อมูล <u>วัน</u> ของตัวแปร <code>birthday</code>
<code>birthday.month</code>	อ้างถึงข้อมูล <u>เดือน</u> ของตัวแปร <code>birthday</code>

19

การกำหนดค่าให้ตัวแปรชนิดโครงสร้าง

- ตัวแปรชนิดโครงสร้างจะมีสมาชิกหลายตัว เราจะต้องกำหนดข้อมูลให้แต่ละสมาชิกในโครงสร้างนั้นๆ ถ้าสมาชิกไม่ได้เป็นข้อความ ให้มองเหมือนการกำหนดตัวแปรปกติ เพียงแต่เราต้องอ้างถึงโดยใช้ชื่อตัวแปรนั้นๆ ก่อน

ตัวอย่าง

```
birthday.day = 30;  
birthday.month = 9;  
birthday.year = 2525;
```

20

ตัวอย่างโปรแกรมการกำหนดตัวแปรโครงสร้าง

```
#include <stdio.h>

struct income {
    float salary,bonus;
    int age;
};

void main(void)
{
    struct income somsri;
    somsri.salary = 18000.00;
    somsri.bonus = 30000.00;
    somsri.age = 23;
}
```

21

การกำหนดสมาชิกของโครงสร้างที่เป็นข้อความ

- ถ้าสมาชิกของโครงสร้างเป็นข้อความ เราจะกำหนด
ตรงๆ เหมือนสมาชิกที่เป็นชนิดอื่น ไม่ได้
- วิธีการกำหนดข้อมูลชนิดข้อความ ให้ทำการกำหนด
ผ่านคำสั่ง **strcpy** หรือรับค่าจากคีย์บอร์ดด้วยคำสั่ง
gets หรือ **scanf** ก็ได้

22

การแสดงผลสมาชิกของตัวแปรชนิดโครงสร้าง

- เนื่องจากสมาชิกในตัวแปรชนิดโครงสร้างก็คือตัวแปรชนิดพื้นฐานทั่วไปในภาษา C ดังนั้นการแสดงผลจะใช้คำสั่ง **printf** และตามด้วยรหัสควบคุมการแสดงผลให้ตรงกับชนิดข้อมูล

23

ตัวอย่างการกำหนดสมาชิกของโครงสร้างที่เป็นข้อความ

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

struct letter {
    char name[20];
    char address[30];
    char message[40];
};

void main(void)
{
    struct letter first;
    printf("Enter name :");
    gets(first.name);
    printf("Enter address :");
    scanf("%s",first.address);
    strcpy(first.message, "How r u?");

    printf("\nNAME is %s",first.name);
    printf("\nAddress is %s",first.address);
    printf("\nMessage : %s",first.message);
}
```



```
Enter name :Somchai Jaidee
Enter address :Bangkok

NAME is Somchai Jaidee
Address is Bangkok
Message : How r u?
```

24

แถวลำดับของตัวแปรชนิดโครงสร้าง

- ปกติเราจะใช้ข้อมูลชนิดโครงสร้างมากกว่าหนึ่งข้อมูล เช่น โครงสร้างนักเรียนประกอบด้วย รหัส, ชื่อ, เพศ, และอายุ โดยส่วนมากเราจะมีข้อมูลของนักเรียนหลายคน ดังนั้นเราสามารถสร้างตัวแปรของนักเรียนให้เป็น array ของตัวแปรโครงสร้าง เพื่อทำการเก็บข้อมูลแบบโครงสร้างไว้เป็นกลุ่มข้อมูลที่เกี่ยวข้องกัน เช่น โครงสร้างนักเรียน ซึ่งโรงเรียนหนึ่งจะต้องมีนักเรียนหลายคนจะต้องเก็บข้อมูลของนักเรียนหลายๆ คนเพื่อนำมาประมวลผลต่อไป

25

แถวลำดับของโครงสร้าง (Arrays of structures)

- ตัวแปรโครงสร้าง 1 ตัวก็คือข้อมูล 1 รายการ (1 record)
- เช่น กำหนดตัวแปรโครงสร้างชนิด student

```
struct student { char code[9];
                  char name[41];
                  float mid, lab, fin;
                  char grade;
                } stu1;
```
- stu1 เป็นตัวแปรโครงสร้าง สามารถเก็บข้อมูลนักเรียนได้ 1 คน
- ต้องการเก็บข้อมูลนักเรียนหลายๆคน ไม่ต้องประกาศตัวแปรโครงสร้างหลายๆตัว
- สามารถสร้างอะเรย์ของโครงสร้างได้

26

การประกาศอระเย์ของโครงสร้าง

■ มีรูปแบบดังนี้

```
struct ชื่อโครงสร้าง ตัวแปรโครงสร้าง[จำนวนสมาชิก];
```

■ เช่น

```
struct student {  
    char code[9];  
    char name[41];  
    float mid, lab, fin;  
    char grade;  
};  
struct student stu[3];
```

■ ได้ตัวแปรอระเย์ที่มีสมาชิก 3 ตัว (เพื่อเก็บข้อมูล น.ศ 3 คน)

■ สมาชิกแต่ละตัวคือ ตัวแปรโครงสร้าง(structure) ชนิด student 27

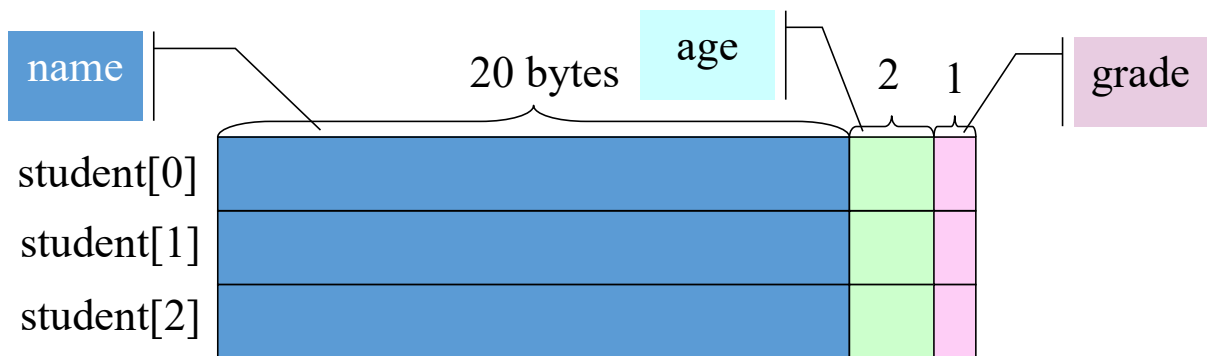
ตัวอย่าง

• ตัวอย่างต่อไปนี้จะแสดงการใช้อระเย์ของโครงสร้าง

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

อะเรย์ของโครงสร้าง

```
struct profile {  
    char name[20];  
    int  age;  
    char grade;  
} student[3];
```



29

การกำหนดค่าเริ่มต้นให้ข้อมูลอะเรย์ของโครงสร้าง

```
struct ชื่อโครงสร้าง ตัวแปร[จำนวนสมาชิก] = {  
    { รายการสมาชิกตัวที่ 1},  
    { รายการสมาชิกตัวที่ 2},  
    ...  
    { รายการสมาชิกตัวที่ n}  
};
```

■ เช่น `struct student stu[100] = { { "61007332", "Somchai", 20, 15, 30, 'C' }, { "61017332", "Decha", 25, 18, 40, 'A' }, { "61027332", "Tanee", 12, 15, 25, 'D' } };`

30

```

#include <stdio.h>
#include <string.h>
main() {
    struct profile {
        char name[20];
        int age;
        char grade;
    }student[10];
    strcpy(student[0].name, "Manee");
    student[2].age = 12;
    student[4].grade = 'A';
    printf("%s\n",student[0].name);
    printf("%d\n",student[2].age);
    printf("%c\n",student[4].grade);
}

```

31

ตัวอย่างโปรแกรม

```

#include <stdio.h>
#include <conio.h>
void main()
{
    struct
    {
        char id[10];
        char name[30];
        int age;
    } std[4];
    int i;

    printf("Please <Enter> for Input 5 record: \n");
    for(i=0;i<4;i++)
    {
        printf("-----\n");
        printf("RECORD # %d \n",i+1);

        printf("\tCODE : ");
        scanf("%s",std[i].id);
    }
}

```

32


```

printf("\tNAME : ");
scanf("%s",std[i].name);

printf("\tAGE:");
scanf("%d",&std[i].age);
}
printf("\n----- OUT PUT ----- \n");
printf(" CODE      NAME      AGE\n");
for(i=0;i<4;i++)
{
    printf("%s      %s      %d \n",std[i].id ,std[i].name,std[i].age);
}
getch();
}

```

33

Please <Enter> for Input 5 record:

RECORD # 1
CODE : 748112
NAME : Piti
AGE:19

RECORD # 2
CODE : 747264
NAME : Somchai
AGE:18

RECORD # 3
CODE : 948264
NAME : Manee
AGE:20

RECORD # 4
CODE : 947497
NAME : Somying
AGE:21

----- OUT PUT -----

CODE	NAME	AGE
748112	Piti	19
747264	Somchai	18
948264	Manee	20
947497	Somying	21

34

จากตัวแปรโครงสร้างนักเรียนที่สร้างไว้

จงเขียนโปรแกรมเพื่อรับจำนวนนักเรียน และข้อมูลนักเรียนจำนวน n คน ซึ่งข้อมูลประกอบด้วย รหัสนักเรียน ชื่อนักเรียน คะแนนเก็บ คะแนนกลางภาค และ คะแนนปลายภาค

จากนั้นคำนวณคะแนนรวมที่นักเรียนแต่ละคนได้

แล้วแสดงผลข้อมูลของนักเรียนแต่ละคน ซึ่งประกอบด้วย รหัสนักเรียน ชื่อนักเรียน และคะแนนรวม

```
2
34561 Somchai 60 20 20
34574 Daungporn 0 0 0
```

- โปรแกรมเก็บข้อมูลของเพื่อน 5 คน
- กำหนดให้มีโครงสร้างดังรูป
- สามารถรับข้อมูลจากผู้ใช้ได้
- สามารถแสดงผลข้อมูลได้
- กำหนดรูปแบบการแสดงผลเอง
- ฟังก์ชันสำหรับการรับข้อมูล
- ฟังก์ชันสำหรับการแสดงผล
- ฟังก์ชันในการคำนวณอายุจากวันเกิด
- ฟังก์ชันในการรับเพศ แล้วเลือกแสดงข้อมูลเฉพาะเพศนั้น

Friend	
Codes:	07520001
Name:	Piti Rugdee
Nickname:	Pete
Sex:	M
Birthday:	09092009

ยูเนียน (Unions)

- คือ การกำหนดรูปแบบชนิดข้อมูลขึ้นมาใหม่ คล้ายกับตัวแปรโครงสร้าง
- เพียงแต่ตัวแปรร้อยภายในจะใช้พื้นที่หน่วยความจำร่วมกัน
- ดังนั้นถ้ากลุ่มตัวแปรร้อยในยูเนียนแล้ว Compiler จะจัดสรรเนื้อที่ในหน่วยความจำให้เท่ากับข้อมูล(ตัวแปร) ที่มีขนาดใหญ่ที่สุด
- ขนาดของตัวแปรแบบยูเนียน จะเท่ากับขนาดของตัวแปรร้อยที่ใหญ่ที่สุด
- ยูเนียน ถูกใช้เมื่อต้องการเก็บข้อมูลเพียงค่าใดค่าหนึ่ง และไม่ยากประกาศตัวแปรเพิ่ม

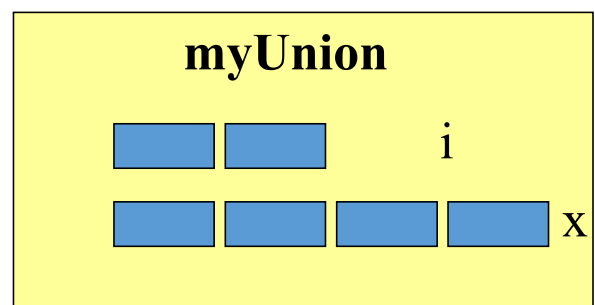
37

ตัวอย่าง

- เช่น

```
union myUnion {  
    int i;  
    float x;  
};  
union myUnion u1;
```

u1 ใช้พื้นที่ขนาด 4 ไบต์



38

- ณ เวลานั้น เราจะสามารถใช้ค่าในตัวแปรแบบยูเนียนได้เพียงตัวเดียวเท่านั้น
- ค่าที่คงอยู่ คือ ค่าของตัวแปรสมาชิกที่ถูกใช้ครั้งล่าสุด เพราะตัวแปรสมาชิกเหล่านั้นจะใช้พื้นที่หน่วยความจำร่วมกัน

เช่น

```
union myunion {
    float i;
    float x;
};
union myunion u1;
u1.i = 5;
printf("%.2f", u1.i);
u1.x = 3.5;
printf("%.2f", u1.x);
printf("%.2f", u1.i);
printf("%.2f", u1.x);
```

ผลการรัน

5.00
3.50
3.50
3.50

39

ตัวชี้ตัวแปรโครงสร้าง

- เราสามารถสร้างตัวแปรพอยน์เตอร์ชนิดโครงสร้างขึ้นมาใช้งานได้ เช่นเดียวกับตัวแปรพอยน์เตอร์ชนิดอื่นๆ เพื่อหาตำแหน่งในหน่วยความจำของโครงสร้างข้อมูล

- รูปแบบการประกาศ

struct ชื่อโครงสร้าง *ชื่อตัวแปรพอยน์เตอร์

```
#include <stdio.h>

main() {
    struct profile {
        char name[20];
        int age;
        char grade;
    };
    struct address *pt;
}
```

```
#include <stdio.h>

main() {
    struct profile {
        char name[20];
        int age;
        char grade;
    } *pt;
}
```

40

การอ้างถึงสมาชิกของตัวแปรโครงสร้างผ่านตัวชี้

- เนื่องจากโครงสร้างประกอบด้วยข้อมูลหลายตัว ดังนั้นเราจึงสามารถอ้างถึงสมาชิกแต่ละตัวในโครงสร้างได้โดยผ่านชื่อพอยน์เตอร์ ด้วยรูปแบบต่อไปนี้

(ชื่อพอยน์เตอร์).ชื่อสมาชิก

ชื่อพอยน์เตอร์->ชื่อสมาชิก

```
#include <stdio.h>

main() {
    struct profile {
        char name[20];
        int age;
        char grade;
    };
    struct address *pt;
}
```

ค่าข้อมูล	ค่าตำแหน่ง
(*pt).name	&(*pt).name
pt->name	&pt->name
(*pt).age	&(*pt).age
pt->age	&pt->age
(*pt).grade	&(*pt).grade
pt->grade	&pt->grade

41

```
#include <stdio.h>

main() {
    struct profile {
        char name[20];
        int age;
        char grade;
    };
    struct address *pt;
    pt->age = 25;
    strcpy(pt->name, "Manee");
    (*pt).grade = 'A';
    printf("&name=%p\n", &(*pt).name);
    printf("name =%s\n", (*pt).name);
    printf("&age =%p\n", &(*pt).age);
    printf("age =%p\n", (*pt).age);
    printf("&grade =%p\n", &pt->grade);
    printf("grade =%c\n", pt->grade);
}
```

42