

สัปดาห์ที่ 4: ข้อมูลแบบโครงสร้าง(Structure)

หัวข้อการเรียนรู้ 1: รูปแบบตัวแปรโครงสร้าง

ความหมาย

คือการรวมกลุ่มของข้อมูลหลายชนิดเข้าด้วยกัน เนื่องจากการจัดเก็บข้อมูลบางครั้งมีข้อมูลหลากหลายรูปแบบเพื่ออธิบายองค์ประกอบของข้อมูล ตัวแปรชนิดที่รวมข้อมูลหลายชนิดเข้าด้วยกันเรียกข้อมูลชนิดนี้ว่า ข้อมูลแบบโครงสร้าง (Structure)

ตัวอย่าง

นักศึกษา 1 คนมีองค์ประกอบข้อมูลหลายอย่าง
เช่น

- ชื่อ
- รหัสนักศึกษา
- เกรดเฉลี่ย
- (หรือ)
- ชื่ออาจารย์ที่ปรึกษา
- วิชาเอก
- ชื่อผู้ปกครอง

จะเห็นว่าข้อมูลภายในประกอบด้วยข้อมูลหลายชนิด ทั้งเหมือนและต่างดังนี้

- ชื่อ เป็นชนิด string
- รหัสนักศึกษา เป็นชนิด int หรือ long
- เกรดเฉลี่ย เป็นชนิด float
- ชื่ออาจารย์ที่ปรึกษา เป็นชนิด string
- วิชาเอก เป็นชนิด string
- ชื่อผู้ปกครอง เป็นชนิด string

ดังนั้นไม่สามารถสร้างตัวแปรชนิดเดียวมารองรับได้

ตัวอย่าง 2

รายวิชาที่เปิดสอน ประกอบด้วยข้อมูลที่สำคัญๆ และ ชนิดข้อมูลดังต่อไปนี้

- รหัสวิชา เป็นชนิด string/ float
- ชื่อวิชา เป็นชนิด string
- อาจารย์ผู้สอน เป็นชนิด string
- จำนวนนักเรียนที่เปิด เป็นชนิด int
- จำนวนนักเรียนที่ลงทะเบียน เป็นชนิด int
- เกรดเฉลี่ยรวม เป็นชนิด float

ทั้งหมดนี้เป็นตัวอย่างเบื้องต้นที่ทำให้เห็นว่าตัวแปรสามารถสร้างข้อมูลย่อยภายใน เก็บข้อมูลได้หลากหลายชนิด

หัวข้อการเรียนรู้ 2: การสร้างตัวแปรโครงสร้าง(Structure)

วิธีการสร้าง

```
struct struct_name
{
    var_type var_name1;
    var_type var_name2;
} ;
```

โดย

- struct_name คือ ชื่อ struct ที่ต้องการสร้าง
- var_type คือชนิดตัวแปรที่ต้องการสร้าง
- var_name คือชื่อตัวแปรที่ต้องการสร้างเป็นองค์ประกอบของตัวแปรหลัก

ตัวอย่าง

จากตัวอย่างก่อนหน้าข้อมูลนักศึกษาประกอบด้วย

- ชื่อ เป็นชนิด string
- รหัสนักศึกษา เป็นชนิด int หรือ long
- เกรดเฉลี่ย เป็นชนิด float
- ชื่ออาจารย์ที่ปรึกษา เป็นชนิด string
- ชื่อผู้ปกครอง เป็นชนิด string

สามารถสร้างตัวแปร ตามที่กำหนดได้ดังนี้

```
struct student
{
    char name[20] ; //ชื่อ
    long id;         //รหัสนักศึกษา
    float gpa ;      // เกรดเฉลี่ย
    char advisor[20]; //ชื่ออาจารย์ที่ปรึกษา
    char guardian[20]; //ชื่อผู้ปกครอง
} ;
```

ตัวอย่าง 2

รายวิชาที่เปิดสอน ประกอบด้วยข้อมูลที่สำคัญๆ และ ชนิดข้อมูลดังต่อไปนี้

- รหัสวิชา เป็นชนิด string/ float
- ชื่อวิชา เป็นชนิด string
- อาจารย์ผู้สอน เป็นชนิด string
- จำนวนนักเรียนที่เปิด เป็นชนิด int
- จำนวนนักเรียนที่ลงทะเบียน เป็นชนิด int
- เกรดเฉลี่ยรวม เป็นชนิด float

สามารถสร้างชนิดตัวแปรโครงสร้างดังนี้

```
struct course
{
    /* รหัสวิชา ชื่อวิชา ชื่อผู้สอน เป็น string */
    char code[8],name[20],teacher[20] ;

    /* จำนวนนักเรียนที่เปิด จำนวนนักเรียนที่ลงทะเบียน เป็น int */
    int capacity, registered;

    /* เกรดเฉลี่ยรวม เป็น float */
    double average_grade ;
};
```

หัวข้อการเรียนรู้ 3: การประกาศโครงสร้าง(Structure)

การประกาศตัวแปร

การประกาศตัวแปร สามารถประกาศได้ดังนี้

```
struct struct_name var;
```

- struct struct_name เป็นชื่อ struct ที่สร้างไว้
- var คือชื่อตัวแปรที่ต้องการเข้าถึง

ตัวอย่าง

```
struct student  
{  
    char name[20] ;  
    long id;  
    float gpa ;  
};
```

ประกาศตัวแปรได้โดย

```
struct student s;
```

โดย s คือตัวแปรชนิด struct student

และตัวแปร s จะประกอบด้วยข้อมูลย่อยคือ name, id และ gpa

การใช้ typedef

คำสั่ง typedef คือคำสั่งที่ใช้สำหรับสร้างตัวแปรใหม่เช่น

typedef int integer;

- typedef คำสั่งสร้างตัวแปรใหม่
- int ชื่อตัวแปรเดิม
- integer ชื่อตัวแปรใหม่

ดังนั้น

เมื่อประกาศ คำสั่ง

typedef int integer แล้ว จะสามารถใช้ integer แทน int ได้เสมอ

เช่น

int i=10;

มีความหมายเช่นเดียวกับ

integer i=10;

การใช้ typedef กับ struct

typedef มักจะใช้กับ struct เพื่อให้การประกาศตัวแปรง่ายขึ้น
จากตัวอย่าง

struct student

{

char name[20] ;

long id;

```
float gpa ;  
};
```

สามารถใช้ typedef สร้างตัวแปรใหม่ เช่น

```
typedef struct student std;
```

- struct student คือชนิดตัวแปรเดิม
- std คือชนิดตัวแปรใหม่

ดังนั้นการประกาศตัวแปรสามารถประกาศตัวแปร s เป็นชนิด struct student ได้สองรูปแบบดังนี้

```
struct student s;   หรือ   std s;
```

หัวข้อการเรียนรู้ 4: การนำเข้าและแสดงผลข้อมูลแบบโครงสร้าง(Structure)

การเข้าถึงข้อมูล

การเข้าถึงข้อมูลชนิดโครงสร้าง สามารถทำได้โดยใช้เครื่องหมาย '.' (จุด) เพื่อเข้าถึงข้อมูลภายในตัวแปร struct

ตัวอย่าง

```
struct student
{
    char name[20] ;
    int id;
    float gpa ;
};
```

หากประกาศตัวแปรโดย กำหนดให้

```
struct student st;
```

จะเข้าถึงข้อมูลภายใน st โดยใช้เครื่องหมายจุด (.) ช่วยดังนี้

เข้าถึงชื่อนักศึกษา ใช้

- st.name

เข้าถึงชื่อรหัสนักศึกษา ใช้



- st.id

เข้าถึงชื่อเกรดเฉลี่ยของนักศึกษา ใช้

- st.gpa

การนำเข้าข้อมูลเข้า

การนำเข้าข้อมูลเข้าหรือแสดงผล ขึ้นอยู่กับชนิดของข้อมูลย่อย ที่เราต้องการตามที่เราระบุไว้ตอนแรก

- st.name  gets(st.name); หรือ
scanf("%s",st.name);
- st.id  scanf("%d",&st.id);




- st.gpa  scanf("%f",&st.gpa);

ตามตัวอย่างจะพบว่า ตัวแปรแต่ละตัว จะเลือกใช้การรับข้อมูลชนิดเดียวกับที่ประกาศไว้ กล่าวคือ

- name เป็น string ก็จะได้รับโดยใช้ %s หรือ gets
- id เป็น int ก็จะได้รับโดยใช้ %d
- gpa เป็น float ก็จะได้รับโดยใช้ %f

การแสดงผลข้อมูล

การแสดงผลข้อมูลก็ทำงานเช่นเดียวกับการรับข้อมูล ชนิดตัวแปรต้องตรงตามกับที่กำหนดไว้ตอนแรก ดังนี้

- st.name  printf("%s\n",st.name);
- st.id  printf("%d\n",&st.id);
- st.gpa  printf("%.2f\n",st.gpa);

// สามารถกำหนดขนาดทศนิยมตามต้องการ

ตัวอย่างการทำงานทั้งหมด

จากตัวอย่าง นำมารวบรวมสร้างโปรแกรม ทั้งรูปแบบรับแสดงผลข้อมูลทั้งหมด

```
void main(){
    /* สร้างตัวแปร typedef */
    typedef struct student std;
    std st;
    /* รับข้อมูล */
    printf("What is student's name?");
```

```
scanf("%s",st.name);
printf("What is student's id?");
scanf("%d",&st.id);
printf("What is student's gpa?");
scanf("%f",&st.gpa);

        /* แสดงผลข้อมูล */
printf("Student information is Name: %s ",st.name);
printf("ID: %d GPA %.2f\n",st.id,st.gpa);
}
```