



การทดลองที่ 6 สตรัคเจอร์ Structure

วัตถุประสงค์

- เพื่อให้เข้าใจการประกาศรูปแบบข้อมูล การประกาศตัวแปรและสามารถใช้งานข้อมูลแบบสตรัคเจอร์ ในภาษา C ได้
- สามารถใช้สตรัคเจอร์ร่วมกับฟังก์ชันได้ โดยการส่งผ่านข้อมูลให้ฟังก์ชัน และส่งค่ากลับจากฟังก์ชันเป็นสตรัคเจอร์
- สามารถใช้สตรัคเจอร์กับอาร์เรย์ได้ ทั้งการใช้อาร์เรย์ภายในสตรัคเจอร์ และการใช้อาร์เรย์ของสตรัคเจอร์ได้

บทนำ

1. การสร้าง structure

การกำหนดโครงสร้างข้อมูลใหม่ขึ้นใช้งานมี 2 ขั้นตอนคือ (1) นิยามโครงสร้างข้อมูล struct และ (2) ประกาศตัวแปรชนิดโครงสร้างข้อมูล struct ที่นิยามขึ้นใหม่ เช่น นิยามโครงสร้างข้อมูล struct student

```
struct student {  
    char name[15];  
    char surname[30];  
    int age;  
};
```

และประกาศตัวแปร std1 และ std2 เป็นแบบชนิดข้อมูล struct student

```
struct student std1, std2;
```

2. การเข้าถึงข้อมูลของตัวแปรแบบ structure

การอ่านค่าและการกำหนดค่าให้กับตัวแปร struct สามารถทำได้โดยการ อ้างถึงชื่อตัวแปร structure ตามด้วย จุด และตามด้วยชื่อสมาชิกที่ต้องการอ่านค่าหรือเขียนค่าข้อมูลเช่น

```
std1.age = 10; // กำหนดให้สมาชิก age ของตัวแปร std1 มีค่าเท่ากับ 10
```

```
strcpy(std1.name, "david"); // กำหนดให้สมาชิก name ของตัวแปร std1 มีค่าเท่ากับ "david"
```

```
printf("%s", std1.name); // พิมพ์ค่าสมาชิก name ของตัวแปร std1 ออกทางจอภาพ
```

3. อาร์เรย์กับสตรัคเจอร์

การเข้าถึงข้อมูลของตัวแปรอาร์เรย์ของสตรัคเจอร์ให้เริ่มจากการอ้างดัชนีของอาร์เรย์ว่าเป็นอาร์เรย์อีลีเมนต์ที่เท่าไร จากนั้นให้อ้างถึงชื่อสมาชิกของสตรัคเจอร์ ที่ต้องการอ่านหรือเขียนข้อมูลเช่น

```
struct date {  
    int day, month, year;  
};  
struct date date_list[3];
```

```
date_list[0].day = 10; // กำหนดให้สมาชิก day ของตัวแปร date_list อีลีเมนต์ที่ 0 มีค่า 10
```

4. ฟังก์ชันและ structure

ตัวแปร structure สามารถเป็นได้ทั้งพารามิเตอร์ของฟังก์ชัน และ เป็นชนิดข้อมูลที่ส่งกลับจากฟังก์ชัน การนิยามค่าของตัวแปรสตรัคเจอร์เป็นอาร์กิวเมนต์ให้กับฟังก์ชันทำได้เหมือนกับตัวแปรชนิดอื่นๆ (int, float, char, double) โดยที่การแก้ไขของตัวแปรสตรัคเจอร์ภายในฟังก์ชัน จะไม่มีผล ต่อค่าของตัวแปรสตรัคเจอร์ที่ถูกส่งมาเป็นอาร์กิวเมนต์ และถ้าในโปรแกรมมีหลายฟังก์ชัน การนิยามสตรัคเจอร์และการกำหนดชนิดตัวแปรใหม่ (typedef) ให้นำมาไว้นอกฟังก์ชัน main



ตอนที่ 1 การประกาศ Structure และ การประกาศชนิดข้อมูลแบบ Structure

1. จงเขียนโปรแกรมในการเก็บรายละเอียดข้อมูลบุคคลโดยกำหนดโครงสร้างข้อมูลบุคคล (person) ให้มีสมาชิกดังนี้
 - ID เก็บรหัสบัตรประชาชนเป็นข้อความจำนวน 13 ตัวอักษร
 - Name เก็บชื่อและนามสกุลมีความยาวไม่เกิน 60 ตัวอักษร
 - Age เก็บอายุเป็นตัวเลขจำนวนเต็ม

ให้ประกาศตัวแปร p1,p2 เป็นตัวแปรแบบโครงสร้างข้อมูล person ให้มีค่าเริ่มต้นคือ p1 มีชื่อว่า "Nadech Naraknaya", อายุ 21 ปี, หมายเลขบัตรประชาชน 3100500511001 และ p2 รับค่าข้อมูลจากผู้ใช้ จากนั้นให้พิมพ์ข้อมูล p1 และ p2 ออกทางจอภาพ ตัวอย่างผลการทำงานของโปรแกรม

```
Enter Name: Mario Maurer
Enter ID: 390980055876
Enter Age: 24
```

```
P1
ID= 3100500511001
Name= Nadech Naraknaya
Age= 21
```

```
P2
ID= 390980055876
Name= Mario Maurer
Age= 24
```

ตอนที่ 2 อาร์เรย์ของสตรัคเจอร์ (Array of structure)

2. จงเขียนโปรแกรมเพื่อเก็บข้อมูลของนักศึกษาจำนวน 4 คน โดยนักศึกษาแต่ละคนให้เก็บข้อมูลดังต่อไปนี้
 - stdcode สำหรับเก็บรหัสนักศึกษาเป็นข้อความจำนวน 10 ตัวอักษร
 - name สำหรับเก็บชื่อและนามสกุลนักศึกษามีความยาวไม่เกิน 60 ตัวอักษร
 - gpa สำหรับเก็บค่าเกรดเฉลี่ยสะสมของนักศึกษา

ให้ประกาศตัวแปร stds เป็น array of structure เพื่อเก็บข้อมูลนักศึกษาทั้ง 4 คนนี้ โดยรับค่าข้อมูลนักศึกษาจากผู้ใช้และสุดท้ายให้พิมพ์ค่าข้อมูลของนักศึกษาทั้ง 4 คนออกจากจอภาพ (ใช้ for loop ช่วยในการรับค่าข้อมูลนักศึกษา)

ตัวอย่างผลการรันโปรแกรม



```

Enter new data
Student1
Code: 10110
Name: David de Gea
GPA: 1.89
Student2
Code: 10112
Name: Patrice Evra
GPA: 3.72
Student3
Code: 10113
Name: Rio Ferdinand
GPA: 1.49
Student4
Code: 10114
Name: Wayne Rooney
GPA: 2.91

All students
10110 David de Gea      GPA= 1.89
10112 Patrice Evra     GPA= 3.72
10113 Rio Ferdinand    GPA= 1.49
10114 Wayne Rooney     GPA= 2.91

```

ตอนที่ 3 สตรัคเจอร์และฟังก์ชัน

3. ให้นักศึกษาปรับปรุงโปรแกรมในข้อที่ 2 โดยเพิ่มฟังก์ชัน showStudent() ที่รับข้อมูลเข้าเป็นโครงสร้างข้อมูลนักศึกษา 1 คน แล้วพิมพ์ค่าข้อมูลของสมาชิกทุกตัวของนักศึกษาคอนนั้นออกทางจอภาพ โดยมีโปรโตไทป์ฟังก์ชันดังนี้

```
void showStudent( struct student st );
```

และให้นักศึกษาปรับปรุงฟังก์ชัน main() เพื่อเรียกใช้งานฟังก์ชัน showStudent() ในการแสดงข้อมูลของนักศึกษาแต่ละคน แทนการเรียกใช้งาน printf() โดยตรงในฟังก์ชัน main() และให้ผลการรันโปรแกรมยังคงเหมือนผลการรันโปรแกรมในข้อที่ 2

หมายเหตุ ในข้อ 3 นี้ให้กำหนดค่าเริ่มต้นให้ตัวแปร stds[4] ได้เลย ไม่จำเป็นต้องวนรับค่าจากผู้ใช้

4. ให้นักศึกษาปรับปรุงโปรแกรมในข้อที่ 3 โดยเพิ่มโครงสร้างข้อมูลนักศึกษาให้มีสมาชิกสำหรับเก็บข้อมูลสถานะของนักศึกษา
- status เก็บสถานะของนักศึกษาโดยมีความยาวข้อความไม่เกิน 15 ตัวอักษร
- และเพิ่มฟังก์ชัน setStatus() เพื่อใช้ในการกำหนดสถานะของนักศึกษา โดยมีโปรโตไทป์ดังนี้

```
struct student setStatus(struct student st );
```

ฟังก์ชัน setStatus() จะคืนค่ากลับเป็นโครงสร้างข้อมูล struct student ที่มีค่า stdcode, name และ gpa เหมือนโครงสร้างข้อมูลนักศึกษาที่รับเข้าไป จากนั้นอ่านค่า gpa เพื่อกำหนดค่าสถานะว่าควรจะมีค่าเป็นเท่าใดตามเกณฑ์ดังนี้

- gpa มากกว่าหรือเท่ากับ 3.50 ให้มีค่าเป็น Excellent
- gpa อยู่ระหว่าง 2.00 ถึง 3.49 ให้มีค่าเป็น Pass
- gpa อยู่ระหว่าง 1.50 ถึง 1.99 ให้มีค่าเป็น Critical
- gpa ต่ำกว่า 1.50 ให้มีค่าเป็น Fail

จากนั้นในฟังก์ชัน main() ให้ปรับปรุงคำสั่งเพื่อเรียกใช้งานฟังก์ชัน setStatus() สำหรับกำหนดค่า status ให้กับนักศึกษาทั้ง 4 คน จากนั้นให้พิมพ์ค่า status ของนักศึกษาทั้ง 4 คนออกทางจอภาพ

**แบบฝึกหัดเสริม**

1. ให้ปรับปรุงฟังก์ชัน showStudent() ให้รับค่าพารามิเตอร์เป็น array ของ struct student เพื่อแสดงค่าข้อมูลของนักศึกษาทุกคนที่อยู่ใน array โดยมีโปรโตไทป์ดังนี้

```
void showStudent( struct student stds[], int n); //n เป็นจำนวนน.ศ.ใน stds[]
```

2. ให้ปรับปรุงฟังก์ชัน setStatus() ให้รับค่าพารามิเตอร์เป็น array ของ struct student เพื่อกำหนดสถานะให้กับนักศึกษาทุกคนที่อยู่ใน array โดยมีโปรโตไทป์ดังนี้

```
void setStatus( struct student stds[], int n); //n เป็นจำนวนน.ศ.ใน stds[]
```

3. จากโปรแกรมในการทดลองตอนที่ 3 ข้อที่ 4 จงเพิ่มฟังก์ชันการทำงาน ต่อไปนี้

- 3.1 ค้นหานักศึกษาที่มี GPA สูงที่สุด ฟังก์ชันคืนค่าสตรัคเจอร์นักศึกษาที่มี GPA สูงที่สุด

```
student maxGPA( struct student arrST[ ] )
```

- 3.2 หาค่าเฉลี่ยของ GPA ของนักศึกษาทั้งหมด (ให้จำนวนนักศึกษาคือ 4 คน)

```
float avgGPA( struct student arrST[ ] )
```

- 3.3 นับจำนวนนักศึกษาที่มีสถานะ (status) "CRITICAL"

```
int countCritical( struct student arrST[ ] )
```