

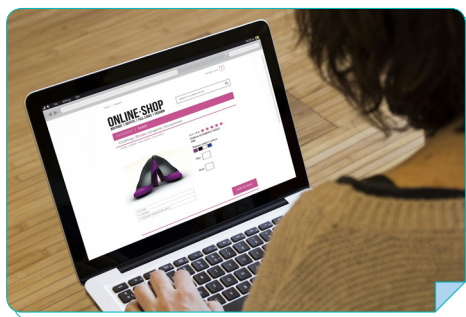
# 8

## 구조체와 공용체

- 학습목표**
- 구조체의 개념을 이해하고, 구조체 사용 방법을 설명할 수 있다.
  - 구조체를 사용하여 프로그램을 작성할 수 있다.
  - 공용체의 개념을 이해하고 공용체를 사용하여 프로그램을 작성할 수 있다.

### 생각 펼치기

인터넷의 발달로 필요한 제품을 인터넷에서 구매할 수 있다. 인터넷에서 제품을 구매할 때 흔히 제품의 정보를 볼 수 있는데, 하나의 제품 정보는 모델명, 무게, 원산지, 크기 등의 다양한 자료로 구성되어 있다. 즉, C 언어의 자료형 중 정수, 실수, 문자 등 다양한 자료형으로 하나의 제품을 설명하는 것이다. 이와 같이 다양한 자료형을 이용하여 하나의 제품을 설명하기 위해서는 여러 자료형을 하나의 변수로 정의할 수 있으면 편리하다.



### 핵심 질문

구조체와 공용체를 사용하여 프로그램을 작성하는 방법은 무엇일까?

### 미션

이 단원을 학습하면서 해결해 보자.



다음은 제품의 상세 정보들을 나타낸 것이다. 이와 같은 정보를 표현할 때 필요한 변수 선언을 하는 프로그램을 작성해 보자(단, 상품 설명은 상품명, 생산자, 가격만 있다고 가정한다.).

#### 자이로 플패키지 세트

배터리+백팩 풀 패키지 세트 구매고객 무료 정품 배터리 1개, 프로그래밍 1개 증정

판매가격 :	1,052,970원	소비자가격 :	1,129,000원
적립금 :	0원	배송비 :	무료배송
상품상태 :	신상품	배송방법 :	택배
구매수량 :	1개	고객선호도 :	★★★★★
제품상태 :	신상품 추천 기획 인기 베스트 새장		
약세사리 :	== 약세사리 선택 ==		

## 1 구조체에 대해 알아보자

배열은 여러 개의 값들을 모아서 저장하고 사용할 때 유용하다. 이때 배열의 값들은 반드시 같은 자료형이어야 한다. 그러나 프로그램 작성 시 종류가 다른 여러 자료형을 하나로 묶어 처리할 때가 있다. 이들 자료형을 하나로 묶어서 한꺼번에 처리할 수 있다면 아주 편리할 것이다. 이러한 경우에 사용하는 것이 **구조체(Structure)**이다.

### 1 구조체 선언

다음 프로그램은 매개 변수로 성명과 3과목의 성적을 전달하면, 그것을 출력하는 사용자 정의 함수를 이용하여 작성한 것이다.

#### 프로그램

```
01 #include <stdio.h>
02 void grade(char *name, int korean, int english, int math)
03 {
04     printf("성명: %s, 국어 %d점, 영어 %d점, 수학%d점\n", name,
05           korean, english, math);
06 }
07 int main( )
08 {
09     grade("강모연", 100, 95, 85);
10     grade("서대영", 87, 77, 84);
11     return 0;
12 }
```

위 함수 자체만 보면 큰 문제가 없어 보인다. 그러나 위와 같은 형태로 함수를 작성하면, 10과목의 성적 처리 시 코드가 복잡해진다.

이 경우 여러 데이터를 하나의 데이터처럼 묶어서 처리할 필요가 있으며, 이때 사용하는 것이 구조체인 struct이다. struct는 구조화된 데이터형을 나타낸다. struct를 사용하면 한 구조체로 선언된 모든 데이터를 가져와 하나의 새로운 데이터형처럼 사용할 수 있다. 구조체를 정의할 때에는 다음과 같이 선언한다.

#### 선언 형식

```
struct 구조체명 {
    자료형 멤버 변수명1;
    자료형 멤버 변수명2;
    ...
};
```

#### 구조체

어떤 객체가 가진 속성들을 모아 놓은 것

#### 실행 결과

성명: 강모연, 국어 100점, 영어 95점, 수학 85점  
성명: 서대영, 국어 87점, 영어 77점, 수학 84점

#### Q & A

**Q** 나만의 자료형을 만드는 것이 가능할까? 예를 들어, 정수형 변수 a를 선언할 때 int a; 라고 하지 않고 jungsu a; 라고 하는 것이 가능할까? 가능하다면 어떤 장점이 있을까?

**A** typedef라는 명령어를 사용하면 가능하다. typedef int stNum;을 프로그램 상단에 입력하면 stNum이라는 정수형 예약어가 정의된다. 따라서 stNum a = 10; 이라는 명령이 가능하다. 물론 변수 a는 stNum형이자 정수형이다. 대형 프로그램을 개발할 때에는 이렇게 형을 새로 만들어 사용하는 경우가 많다. 형 이름에 의미를 두어 프로그램을 읽기 쉽게 하기 위해서이다.

## 2 구조체의 정의

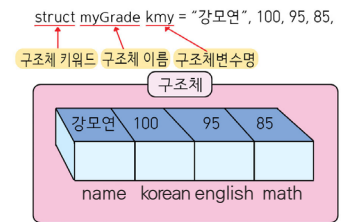
실제 앞에서 사용한 각 학생별 성적 정보 구조체는 다음과 같이 정의할 수 있다.

```
struct myGrade {
    char *name;
    int korean;
    int english;
    int math;
};
```

구조체 키워드와 구조체 이름

구조체 멤버

위 코드는 여러 데이터를 가진 하나의 사용자 정의 데이터형을 새로 만든다. 구조체를 정의한 뒤에 이 구조체를 사용하려면 일반적인 원시 변수를 선언 하듯이 구조체 변수를 선언하고 정의한다. 이때 선언과 정의는 별도로 할 수 있다. 구조체 변수가 생성 되면 그 구조체 안에 있는 각각의 멤버에 접근할 때에는 (.)연산자를 사용한다.



```
printf("%d", kmy.math);
```

**예제** myGrade 구조체를 이용하여 kmy라는 구조체 변수를 선언하고 사용하는 프로그램을 작성해 보자.

### 실행 결과

성명: 강모연, 국어  
100점, 영어 95점, 수  
학 85점

### 프로그램

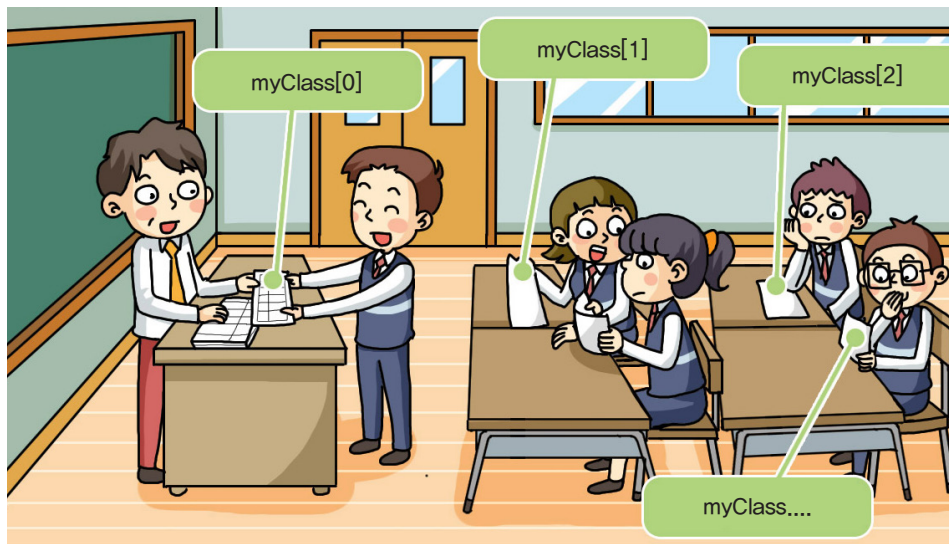
```
01 #include <stdio.h>
02 struct myGrade {
03     char *name;
04     int korean;
05     int english;
06     int math;
07 };
08 void grade(struct myGrade st)
09 {
10     printf("성명: %s, 국어 %d점, 영어 %d점, 수학 %d점\n", st.name,
11           st.korean, st.english, st.math);
12 }
13 int main( )
14 {
15     struct myGrade kmy = {"강모연", 100, 95, 85};
16     grade(kmy);
17     return 0;
18 }
```

구조체 키워드와 구조체 이름

구조체 멤버

## 2 구조체 배열을 만들어 사용하자

구조체 변수는 학생 한 명의 데이터만을 저장할 수 있다. 그러나 한 학급의 구성원이 20명이라면 20명의 성적 정보를 저장해야 한다. 이때 사용하는 것이 **구조체 배열**이다. 20명의 성적 정보를 저장할 때 20개의 구조체를 저장할 구조체 배열을 선언하고, 배열의 각 요소마다 한 사람의 이름과 3과목 성적 구조체를 연결한다.



[그림 II-11] 구조체 배열을 이용한 성적 정보 입력

myGrade 구조체의 배열을 선언하는 형식은 일반적인 배열을 선언하는 것과 다르지 않다. 다만 구조체 배열이므로 앞에 struct를 붙이면 된다.

myGrade를 이용하여 20명의 이름과 성적 정보를 저장할 수 있는 구조체 배열 myClass는 다음과 같이 선언한다.

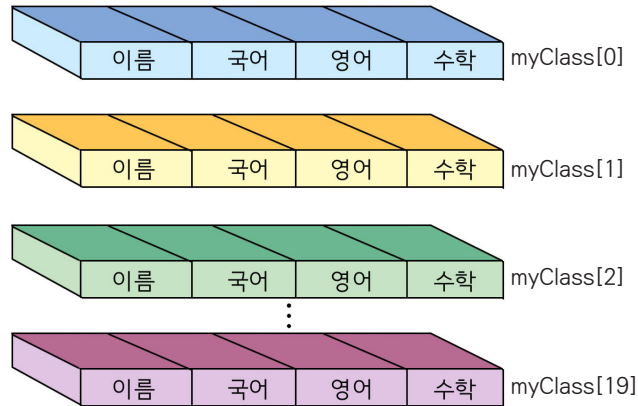
```
struct myGrade {
    char *name;
    int korean;
    int english;
    int math;
};
int main(void)
{
    struct myGrade myClass[20];
    ...
}
```

20명의 성적 정보 저장

### Q & A

**Q** 배열과 포인터는 매우 유사하다고 하였다. 그러면 구조체 포인터를 만드는 것도 가능할까?

**A** 구조체 포인터를 사용하면 배열이 가진 기억 장소 고정 할당의 단점을 해결할 수 있다. 배열은 선언할 때 크기가 고정되지만, 포인터를 이용하면 필요할 때마다 기억 장소를 할당받아 사용할 수 있다. 이런 장점으로 구조체 포인터는 컴퓨터의 자료 구조(리스트, 스택, 큐 등)를 구현할 때 사용한다.



[그림 II - 12] myClass 구조체 배열 저장 구조

구조체 배열을 사용하여 각 학생의 성명과 성적 정보를 저장할 때에는 다음 프로그램에서와 같이 배열 첨자와 도트(.) 연산자를 같이 사용하면 구조체 멤버(변수)에 접근할 수 있다.

#### 프로그램

```

01  #include <stdio.h>
02  struct myGrade {
03      char *name;
04      int korean;
05      int english;
06      int math;
07  };
08  int main(void)
09  {
10      struct myGrade myClass[20];
11      myClass[0].name = "강모연";
12      myClass[1].name = "서대영";
13      myClass[0].korean = 100;
14      myClass[1].math = 84;
15      printf("%s의 국어 점수는 %d\n", myClass[0].name, myClass[0].korean);
16      printf("%s의 수학 점수는 %d\n", myClass[1].name, myClass[1].math);
17      return 0;
18  }

```

구조체 배열도 초기화가 가능하므로 구조체 배열을 선언할 때 바로 초기화할 수 있어요. 이 경우에는 중괄호 안에 또 중괄호를 사용해야 돼요.

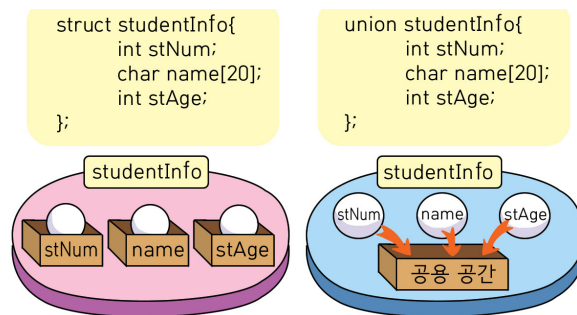


#### 실행 결과

강모연의 국어 점수는 100  
서대영의 수학 점수는 84

### 3 공용체에 대해 알아보자

C 언어에는 하나의 메모리 공간을 여러 개의 변수들이 공유할 수 있게 하는 기능이 있다. 이것을 **공용체(Union)**이라 하며, 하나의 메모리 공간을 여러 개의 변수가 공유하는 이유는 메모리를 절약하기 위해서이다. 공용체는 구조체와 비슷하지만 구조체는 구조체 안의 멤버 변수가 하나의 메모리 공간을 공유하지는 않는다.



[그림 II-13] 구조체와 공용체

다음은 공용체를 이용하여 메모리의 크기를 알아보는 프로그램이다.

#### 프로그램

```
01  #include <stdio.h>
02  union studentInfoUnion {
03      int stNum;
04      char name[20];
05      int stAge;
06  };
07  int main(void)
08  {
09      printf("%d byte", sizeof(union studentInfoUnion));
10      return 0;
11  }
```

#### 실행 결과

20byte

위 프로그램을 실행하면 공용체 studentInfoUnion에 가장 큰 기억 공간을 차지하는 name[20]의 크기가 출력된다. 즉 20 byte가 출력된다.

#### Q & A

**Q** 만약 옆 코드에서 Union 키워드를 struct로 바꾸면 몇 byte가 출력될까? 구조체와 다르게 출력된다면 왜 그럴까?

**A** 다르게 출력된다. 공용체는 멤버 변수가 메모리를 공유하기 때문에 가장 큰 변수의 기억 공간의 크기가 공용체의 크기가 되지만, 구조체는 멤버 각각의 기억 공간을 합한 것이 구조체의 크기가 된다. 따라서 Union 키워드를 struct로 바꾸면 28byte가 출력된다.

## 예제

구조체를 이용하여 거래처 전화 번호부를 만들어 보자(단, 전화번호부에는 성명, 회사명, 전화번호 등이 저장되어야 한다.).

### 조건

- ① 성명, 회사명, 전화번호를 저장하며, 모두 문자형 배열로 한다.
- ② gets( ) 함수를 이용하여 문자를 받는다.
- ③ 구조체의 이름은 contacts라고 한다.
- ④ 구조체 배열을 이용하여 최대 20개의 거래처를 등록할 수 있다.

### 프로그램

```
01 #include <stdio.h>
02 #include <string.h>
03 struct contacts {
04     char name[10];
05     char office[20];
06     char phone[15];
07 };
08 int main(void)
09 {
10     struct contacts customer[20];
11     printf("성명을 입력하시오: ");
12     gets(customer[0].name);
13     printf("회사명을 입력하시오: ");
14     gets(customer[0].office);
15     printf("전화번호를 입력하시오: ");
16     gets(customer[0].phone);
17     printf("\n");
18     printf("성명: %s \n", customer[0].name);
19     printf("회사명: %s \n", customer[0].office);
20     printf("전화번호: %s \n", customer[0].phone);
21     return 0;
22 }
```

메모리 블록이나 문자열을 다룰 수 있는 함수 포함.

### 실행 결과

성명을 입력하시오: 홍길동  
회사명을 입력하시오: 한국로봇시스템  
전화번호를 입력하시오: 070-1234-1234

성명: 홍길동  
회사명: 한국로봇시스템  
전화번호: 070-1234-1234

## 미션

### 해결하기

인터넷 쇼핑을 하면 하나의 제품을 설명하는 자세한 정보들이 있다. 정보들을 표현할 때 필요한 구조체 변수 선언을 하는 프로그램을 작성해 보자(단, 상품 설명은 상품명, 생산지, 가격만 있다고 가정한다.).

#### 프로그램

```
01 struct product {
02     char productName[50];
03     char madein[50];
04     int price;
05 };
```

## 스스로 해결하기

직원 정보를 저장하는 프로그램을 작성하려고 한다. 직원 5명의 정보를 입력할 수 있는 구조체 배열을 선언하자. 5명의 정보를 넣은 후 이름으로 권순동을 입력받고, 권순동의 사원 번호와 전화 번호를 출력하는 프로그램을 작성해 보자.

사원 번호	성명	전화 번호
s2001002	강대영	010-1234-0001
s2006012	김길수	010-1234-0002
s2011007	이미영	010-1234-0003
s2012005	이도엽	010-1234-0004
s2016001	권순동	010-1234-0005

#### 알고리즘 설계

- ① 5명의 직원 정보는 구조체 배열을 이용하여 프로그램에서 초기화하여 입력한다.
- ② 권순동은 scanf( ) 함수를 이용하여 입력받는다.

#### 프로그램





## 수행 평가

## 구조체를 멤버로 가지는 구조체

**활동 목표** 구조체를 멤버로 가지는 구조체를 선언하고 사용할 수 있다.

다음 프로그램에서 human 구조체는 person 구조체를 이용하여 친구 정보를 저장하고 있다.

프로그램이 실행되도록 빈칸에 코드를 채워 프로그램을 완성해 보자.

### 프로그램

```
01  #include <stdio.h>
02  struct person {
03      int age;
04      char name[10];
05  };
06  struct human {
07      int age;
08      char name[10];
09      
10  };
11  int main( )
12  {
13      struct human man1 = {20, "서대영", 20, "유시진"};
14      printf("%s의 친구 이름은 %s입니다.", man1.name, man1.myfriend.name );
15      return 0;
16  }
```

### 스스로 평가하기

평가 항목	구분		
	그렇다	보통이다	그렇지 않다
• 구조체의 기능과 장점을 설명할 수 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
• 구조체를 이용하여 프로그램을 작성할 수 있다.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



## 내 실력 확인하기

### 내용을 이해했나요?

- **구조체**: 서로 다른 종류의 데이터를 묶어서 한 개의 변수처럼 사용하기 위한 것이다.
- **구조체 멤버**: 구조체에 포함된 각각의 변수들이다.
- **구조체 배열**: 구조체를 여러 개 생성하기 위해 배열을 이용하여 구조체를 선언한 것이다.
- **공용체**: 구조체와 비슷한 형태를 가지지만 멤버들이 하나의 기억 공간을 공유하며 사용하는 것이다.

### 문제로 확인할까요?

#### 1. 구조체를 정의하는 키워드로 옳은 것은?

- ① structure      ② union      ③ struct      ④ cast      ⑤ typedef

#### 2. 다음의 구조체 student가 차지하는 메모리 공간은 몇 byte인가?

##### 프로그램

```

01  #include <stdio.h>
02  struct student {
03      int age;
04      char name[10];
05  };
07  int main(void)
08  {
09      struct student s1 = {17, "홍길동"};
10      printf("%s학생의 나이는 %d입니다. ", s1.name, s1.age);
11      return 0;
12  }
    
```

##### 실행 결과

### 평가해 볼까요?

★ 다음 평가 항목에 따라 자신의 성취 척도를 스스로 점검해 보자.

영역	평가 항목	척도				
		1	2	3	4	5
이해	구조체와 공용체의 개념과 구조를 이해하고, 구조체 선언과 초기화 방법을 설명할 수 있는가?					
적용	구조체 및 구조체 배열을 사용한 프로그램을 작성할 수 있는가?					