

# 고급C프로그래밍

09 구조체(2/2)

# 중간고사 review – 10/11번 문제

- 10번 문제 : 잘못된 부분을 찾고 이유를 기술

```
.....  
int *aa[3];  
  
for(int bb=0; bb<3; bb++)  
    aa[bb] = bb;  
  
for(int cc=0; cc<3; cc++)  
    printf("%d", *aa[cc]);  
.....
```

aa는 포인터 배열  
→ 주솟값을 저장해야 함

# 중간고사 review – 10/11번 문제

## • 11번 문제

- ✓ 1안) 포인터 배열 aa에 주솟값을 직접 대입

→  $aa[bb] = \&bb;$

- ✓ 이 경우 컴파일 에러는 발생하지 않으나, 출력되는 값은 333

→ bb의 주소는 하나이며, 따라서 aa[bb]에 동일한 주소가 저장되고, 해당 주소에 저장되는 값은 bb의 최종값인 3임

- ✓ 2안) 포인터 배열 aa에 동적 메모리를 할당하고, 해당 메모리에 bb의 값을 대입, 저장/출력되는 값은 012

→  $aa[bb] = (\text{int}^*)\text{malloc}(\text{sizeof}(\text{int}));$

$*aa[bb] = bb;$

## 중간고사 review – 13번 문제

- 사용자로부터 하나 이상의 복수개의 문자열을 입력받고, 그 내용을 출력하는 프로그램
  - 사용자가 “end”를 입력하면 더 이상의 문자열 입력은 없는 것으로 판단
  - “end”는 입력받은 문자열에 포함
  - 동적 메모리 할당과 포인터 배열을 사용하여 (1)~(4)번 빈칸의 코드를 완성

```
.....  
void main( ){  
char ** input_strings; // 문자열들을 저장하기 위한 포인터 배열  
char buffer[10];  
int count = 0;  
input_strings = (char **)malloc(sizeof(char *));
```

## 중간고사 review – 13번 문제

```
while(1){  
    printf("문자열을 입력하세요 (종료하려면 'end' 입력): ");  
    gets_s(buffer, 10); // 문자열을 입력받음  
    input_strings[count] = (char *)malloc(sizeof(char) * (strlen(buffer)  
+ 1)); //입력받은 문자열을 저장하기 위한 동적 메모리 할당  
    strcpy(input_strings[count], buffer); // 문자열 복사  
    puts(input_strings[count]); // 입력받은 문자열을 출력  
    if(strcmp(input_strings[count], "end") != 0) {  
        input_strings = (char **)realloc(input_strings, sizeof(char *)  
* (count + 2)); //포인터 배열의 크기를 한 칸 늘려줌  
        count++;    }  
    else break;    }
```

## 중간고사 review – 13번 문제

```
for(int a = 0; a <= count; a++)
```

**free(input\_strings[a]); //각 포인터 배열의 행별로 할당받은  
메모리를 반환**

```
free(input_strings); } //포인터 배열의 메모리를 반환
```

# 실습3 review

- 학생들의 숫자, 각 학생 별 이름, 국어/영어 과목 점수를 사용자로부터 입력 받고 평균 점수를 저장하는 프로그램
  - ✓ 각 학생들의 이름과 평균 점수, 과목별 평균 점수 그리고 모든 학생들의 총 평균 점수를 average\_student.txt에 출력
  - ✓ 학생들의 이름, 국어 점수, 영어 점수는 구조체로 저장/관리하며 동적 할당
  - ✓ 각 학생들의 평균 점수와 과목별 평균 점수, 모든 학생들의 총 평균 점수는 구조체로 저장/관리
- 고려사항 : 각 학생들의 평균 점수를 어떻게 구조체의 멤버로 정의할 것인가?
  - ✓ 각 학생들의 평균 점수 개수는 프로그램이 실행될 때마다 가변적임
  - 동적 메모리 할당
  - ✓ 각 학생들의 평균 점수를 어떻게 저장할 것인가?
  - 1안) 이름과 평균 점수로 구성된 구조체를 생성, 동적 메모리 할당을 통한 구조체 배열로 저장.. 그 외 여러가지 방안을 고려해 볼 수 있음

## (2) 공용체

# 공용체의 개념

- 하나의 공간(메모리)을 서로 다른 두 변수가 같이 사용하는 것

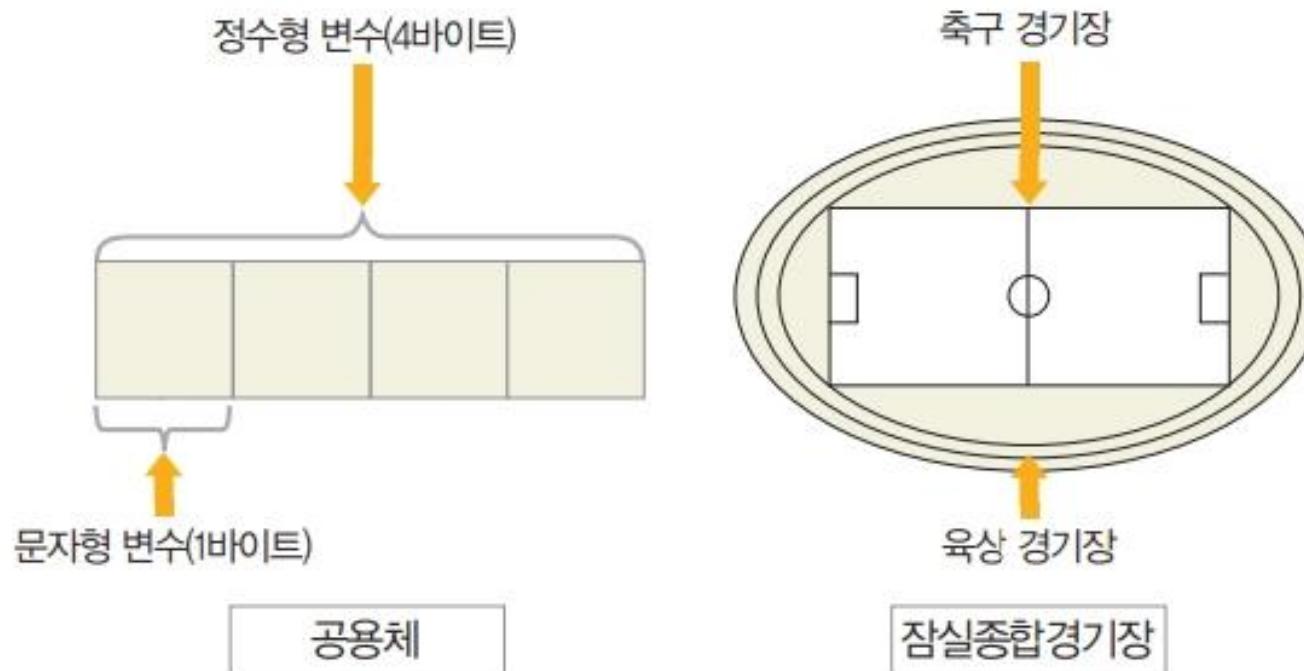


그림 13-7 공용체와 잠실종합경기장의 개념 비교

# 공용체의 문법(1/4)

- 구조체와 유사, struct 대신 union을 사용

- ✓ 구조체 : 멤버 변수 각각에 별도의 메모리 공간을 할당
- ✓ 공용체 : 멤버 변수가 공간을 공유

```
union 공용체형_이름 {  
    데이터_형식 멤버_변수_1;  
    데이터_형식 멤버_변수_2;  
    :  
};  
  
union 공용체형_이름 공용체_변수;
```

# 공용체의 문법(2/4)

## 기본 13-7 공용체 사용 예

13-7.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     union student {
06         int tot;
07         char grade;
08     };
09
10     union student u;
11
12     u.tot = 300;
13     u.grade = 'A';
14
15     printf("\n— 공용체 활용 —\n");
16     printf("총점 ==> %d\n", u.tot);
17     printf("등급 ==> %c\n", u.grade);
18 }
```

—— 공용체 변수 student를 선언한다.

실행 결과

4 --- 공용체 활용 ---  
총점 => 321  
등급 => A

—— 공용체 변수 u를 선언한다.

—— 공용체 변수의 멤버 변수에 값을 대입한다.

—— 공용체 변수의 멤버 변수값을 출력한다.

# 공용체의 문법(3/4)

- 5행에서 union 예약어를 사용하여 공용체를 선언, 10행에서 공용체 변수 선언

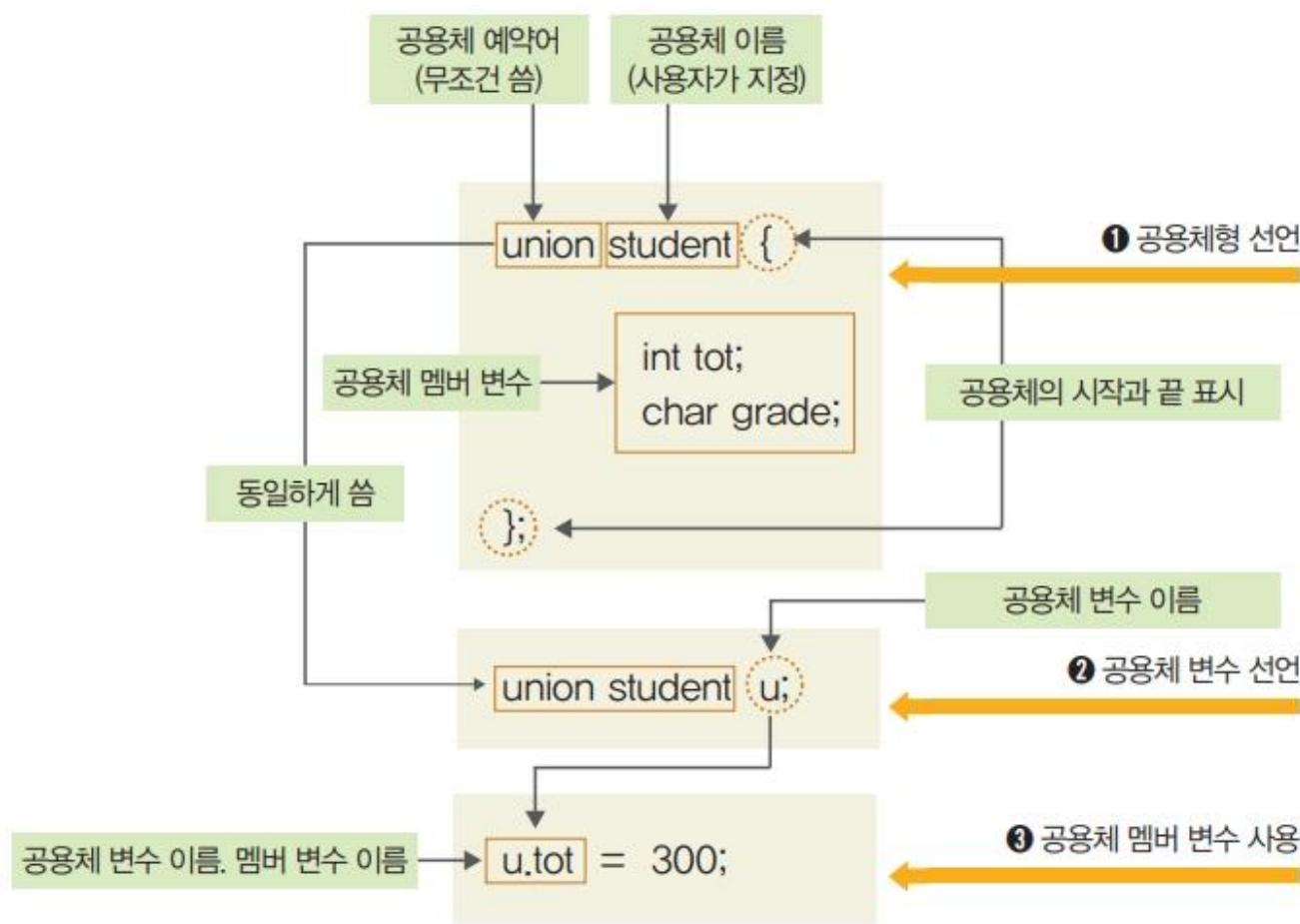


그림 13-8 공용체의 문법 구조

# 공용체의 문법(4/4)

- 공용체에서 데이터가 저장되는 방식

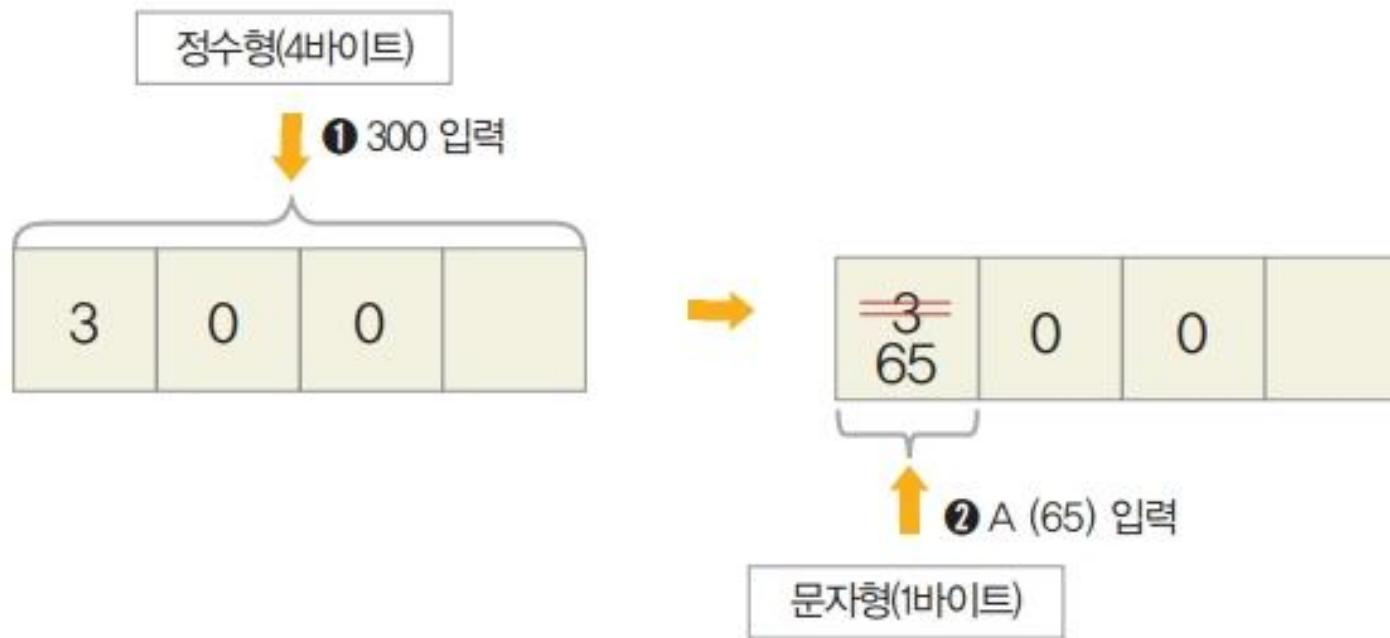


그림 13-9 [기본 13-7]의 저장 과정

## 여기서 잠깐 공용체의 크기

공용체 멤버 변수가 여러 개일 때는 그중 가장 큰 저장 공간이 공용체의 크기가 됨

# 공용체를 언제 사용할까?

- 작은 메모리 공간으로 다양한 데이터 타입의 값을 저장할 때
- 멤버들중 조건에 따라 하나만 값을 할당받을 때
- 멤버의 값의 변화가 다른 멤버에게 영향을 끼치도록 할 때

# (3) 열거형

# 열거형 문법 (1/3)

- 연속된 정수값에 이름을 붙여서 표현 가능

- ✓ 요일을 열거형으로 표현
- ✓ 기본적으로 나열한 데이터의 값은 0부터 1씩 차례대로 증가
- ✓ 명시적으로 값을 지정할 수도 있음  
→ sun = 1, mon = 3, .....  
→ 값이 지정되지 않은 항목은 이전 항목의 값부터 1씩 증가



# 열거형 문법 (2/3)

## 기본 13-8 열거형 사용 예

13-8.c

```
01 #include <stdio.h>
02
03 void main( )
04 {
05     enum week {sun, mon, tue, wed, thu, fri, sat};      ---- 0부터 6까지의 열거형이다.
06
07     enum week ww;                                     ---- 열거형 변수 ww를 선언한다.
08
09     ww = sat;                                         ---- 변수 ww에 값을 대입한다.
10
11    if(ww == sun)                                     ---- ww가 sun(0)인지 아닌지의
12        printf("오늘은 일요일입니다.\n");             여부에 따라 문장을 출력한다.
13    else
14        printf("오늘은 일요일이 아닙니다.\n");
15 }
```

실행 결과

오늘은 일요일이 아닙니다.

## 열거형 문법 (3/3)

- 9행에서 ww에 sat(실제 값은 6)를 대입, 11행에서 ww의 값에 따라 일요일인지 여부를 출력

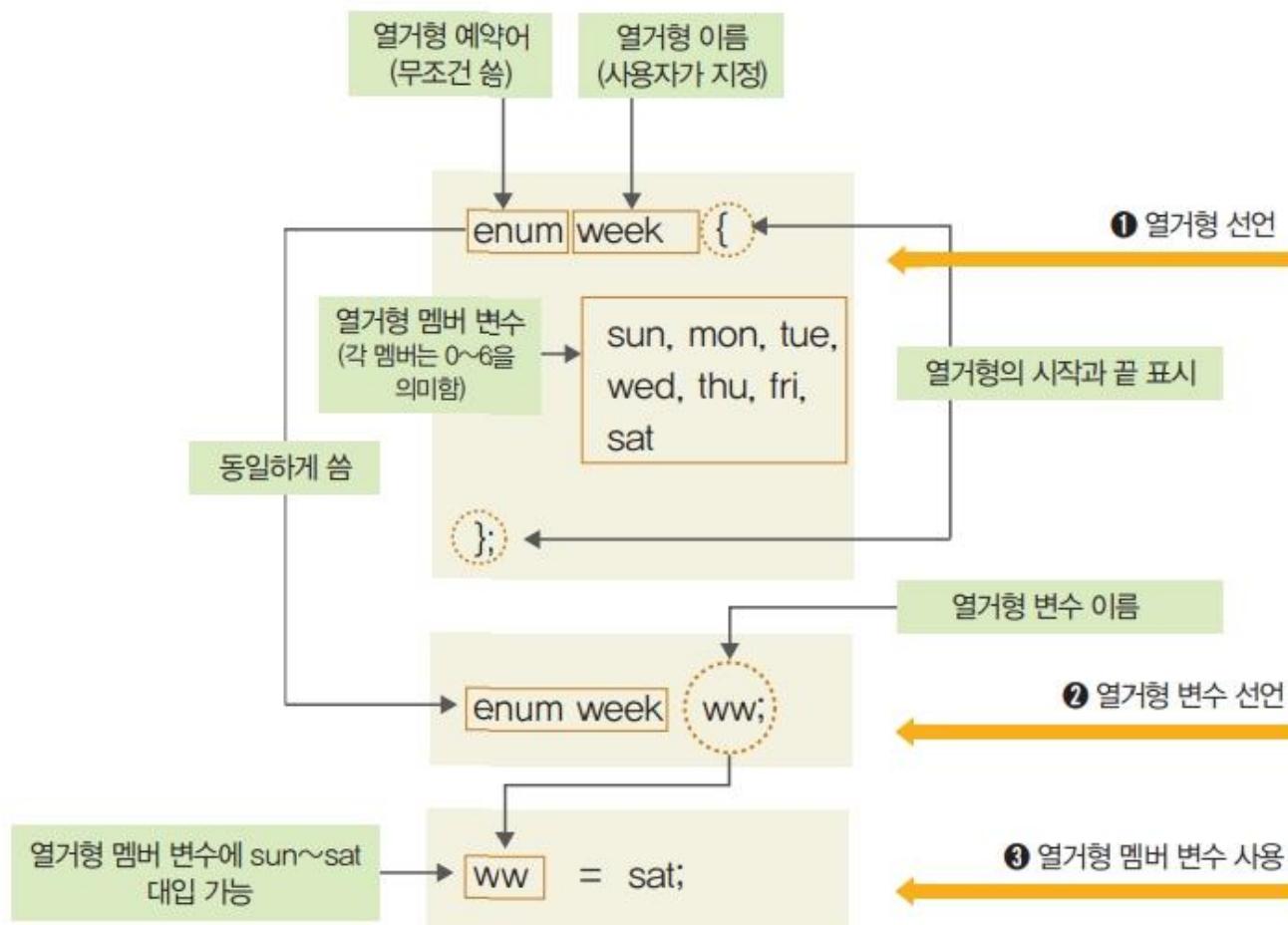


그림 13-11 열거형의 문법 구조

실습

# [실습 1] 이름과 전화번호(또는 주민번호) 저장

**예제 설명** 사용자 이름을 입력하고 전화번호 또는 주민번호 중 한 가지만 입력하는 프로그램이다. 전화번호와 주민번호는 공용체로 동일한 메모리를 차지함으로써 공간을 절약한다. 구조체와 공용체를 혼합해서 사용해보자.

## 실행 결과

이름 --> 채영  
전화번호 또는 주민번호 --> 990423-1234567

--- 구조체/공용체 혼합 활용 ---

이름 => 채영  
전화번호/주민번호 => 990423-1234567

## • 구조체의 멤버로 공용체 변수를 선언

- ✓ 이름과 (전화번호 또는 주민번호)를 저장하는 구조체를 선언
- ✓ 전화번호와 주민번호는 문자열로 저장

# [실습 1] 이름과 전화번호(또는 주민번호) 저장

- 구조체의 멤버로 공용체(또는 구조체) 변수를 선언할 경우

- ✓ 구조체 선언 시 멤버 변수명(공용체 또는 구조체)을 정해줘야 함

- 멤버 변수로 생성할 공용체 또는 구조체를 미리 선언하는 방법

- 선언과 동시에 변수를 생성하는 방식을 활용할 수도 있음

① 구조체형과 변수를 별도로 선언

```
struct student {  
    char name[10];  
    int kor;  
    int eng;  
    float avg;  
};
```

**struct student s;**

② 구조체형과 변수를 동시에 선언

```
struct student {  
    char name[10];  
    int kor;  
    int eng;  
    float avg;  
}s;
```

③ typedef를 이용하여 선언

```
typedef struct _student {  
    char name[10];  
    int kor;  
    int eng;  
    float avg;  
}student;  
  
student s;
```

## [실습 2] 입력된 월의 이름을 출력

**예제 설명** 열거형을 활용해서 입력된 월의 이름을 출력하는 프로그램이다.

**실행 결과**

월 입력 : 8

8월은 August 입니다.

- 모든 월을 나타내는 열거형을 생성
  - ✓ January는 1로 설정
- switch 문을 이용해서 월의 이름을 출력

## [실습 3] 도서관 도서 대여 프로그램

- **도서관에서는 보유 도서에 대해 다음과 같은 정보를 이용해서 관리함**
  - ✓ 도서 ID, 도서 제목, 보유 권수
  - ✓ 보유 도서명과 보유 권수는 book.txt에 기술되어 있음
- **도서관에서는 현재 보유 도서 현황을 다음과 같은 정보를 이용해서 관리함**
  - ✓ 보유 도서 정보 리스트
- **도서관에서는 대여 도서 현황을 다음과 같은 정보를 이용해서 관리함**
  - ✓ 대여 도서 정보 리스트

## [실습 3] 도서관 도서 대여 프로그램

- **프로그램의 동작**

1. 사용자에게 현재 보유한 도서 정보 리스트를 제공
2. 사용자로부터 대여를 원하는 도서의 ID를 입력 받음
  - 2.1. 대여 가능할 경우 도서 대여 메시지를 출력
  - 2.2. 보유 도서 정보 리스트에 없는 도서의 ID를 입력 받을 경우
    - 2.2.1 보유 도서가 모두 대여되었을 경우 관련 메시지 출력
    - 2.2.2 보유하지 않은 도서일 경우 관련 메시지 출력

## [실습 3] 도서관 도서 대여 프로그램

- **프로그램의 구성**

- ✓ `main()` : 사용자로부터 도서 대여 요청을 받고 결과를 출력함
- ✓ `initial_booklist()` : 보유 도서 현황을 초기화 함
- ✓ `manage_booklist()` : 사용자 요청에 따라 보유 도서 현황(또는 대여 도서 현황)을 업데이트 함

# Q & A