



AJOU UNIVERSITY



# 컴퓨터 프로그래밍 및 실습

11주차. 구조체

# 실습 안내

## ■ 실습 제출 안내

- 솔루션 이름은 "Practice week 11"
- 프로젝트 이름과 소스코드 이름은 Problem1, Problem2, ...
  - 실습1의 프로젝트 이름은 Problem1, 소스코드 이름은 problem1.c
  - 실습 2의 프로젝트 이름은 Problem2, 소스코드 이름은 problem2.c ...
- 솔루션 폴더를 압축하여 **Practice\_week11\_학번\_이름.zip** 으로 제출
- 제출기한: 당일 **19시** 까지

# 1. 구조체의 사용

## ■ 구조체 사용의 이유

- 학생 정보를 배열에 담아 저장하고자 한다.
- 학생 정보는 (학번, 이름, 학점) 튜플이다.
- 위 세 정보를 한꺼번에 담을 수 있는 방법은 없을까?

## ■ 만약 구조체가 없다면...

- `int student_id[10] = { 2022135486, 20162351, ... }`
- `char* student_id[10] = { "홍길동", "박재상", ... }`
- ...
- 굉장히 비효율적
  - 학생에 대한 데이터를 하나로 모으려면?



학번: 20100001(정수)  
이름: "최자영"(문자열)  
학점: 4.3(실수)  
...

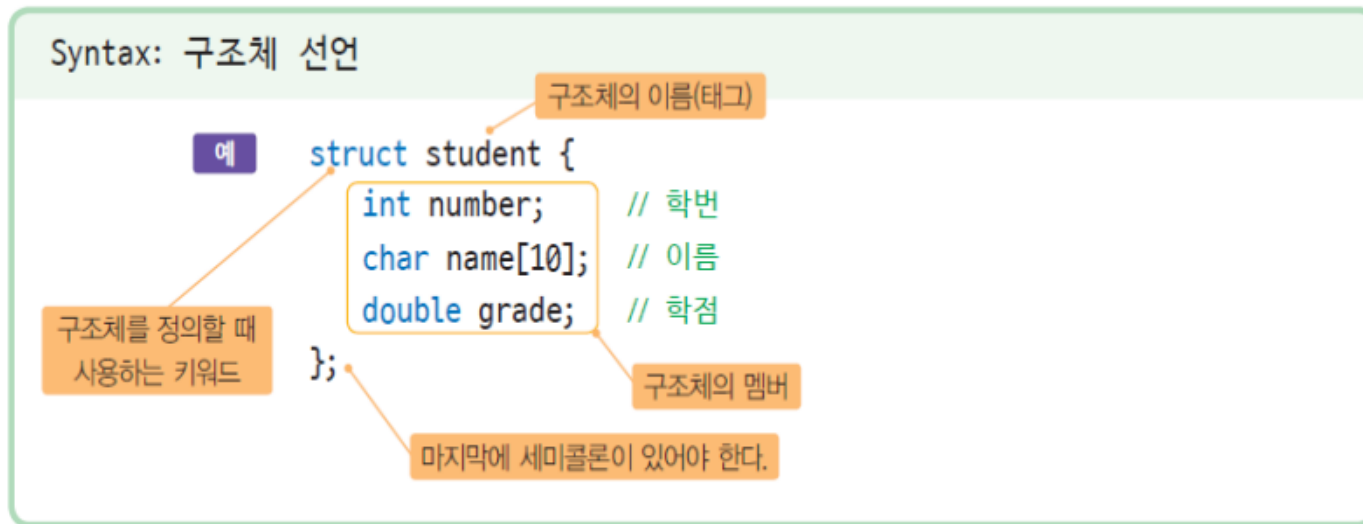
`int number;`  
`char name[10];`  
`double grade;`  
와 같이 개별 변수  
로 나타낼 수 있지  
만 묶을 수가 있나?

# 1. 구조체의 사용

## ■ 구조체의 선언

- 사용자가 새로운 타입을 만들어내는 것과 비슷
- number, name, grade를 담을 수 있는 하나의 타입을 선언한 것

### ❖ 구조체 선언



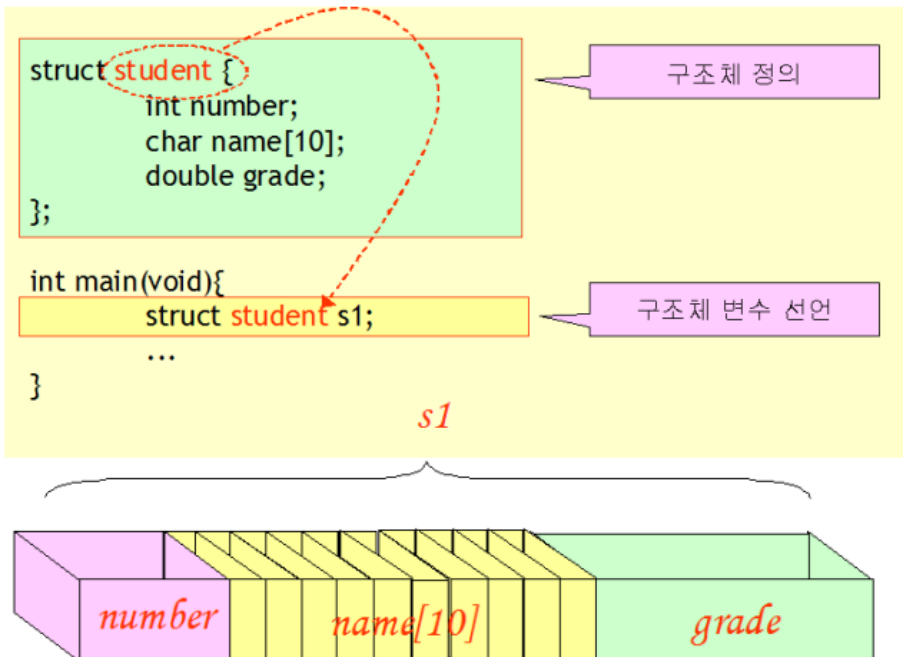
### ❖ 구조체 선언은 변수 선언이 아님

# 1. 구조체의 사용

## ■ 구조체의 선언

- 사용자가 새로운 타입을 만들어내는 것과 비슷
- number, name, grade를 담을 수 있는 하나의 타입을 선언한 것

### ❖ 구조체 변수 선언

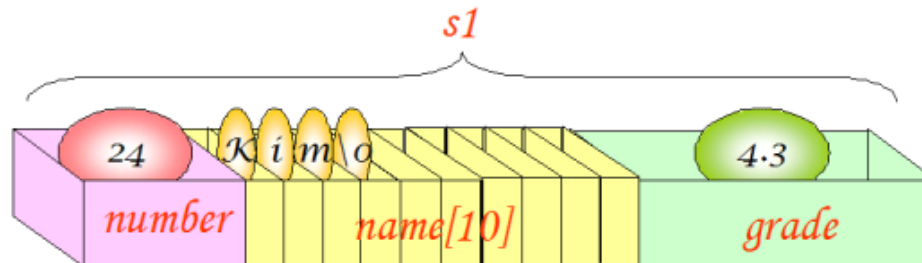


# 1. 구조체의 사용

## ■ 구조체의 선언

- 사용자가 새로운 타입을 만들어내는 것과 비슷
- number, name, grade를 담을 수 있는 하나의 타입을 선언한 것
  - ❖ 구조체의 초기화

```
struct student {  
    int number;  
    char name[10];  
    double grade;  
};  
struct student s1 = { 24, "Kim", 4.3 };
```



# 1. 구조체의 사용

## ■ 구조체 멤버에 대한 접근

### ❖ 예시

```
struct student {  
    int number;  
    char name[10];  
    double grade;  
};  
  
int main(void)  
{  
    struct student s;  
  
    printf("학번을 입력하시오: ");  
    scanf("%d", &s.number);  
  
    printf("이름을 입력하시오: ");  
    scanf("%s", s.name);  
  
    printf("학점을 입력하시오(실수): ");  
    scanf("%lf", &s.grade);  
  
    printf("\n학번: %d\n", s.number);  
    printf("이름: %s\n", s.name);  
    printf("학점: %f\n", s.grade);  
    return 0;  
}
```

학번을 입력하시오: 20190001  
이름을 입력하시오: 홍길동  
학점을 입력하시오(실수): 4.3

학번: 20190001  
이름: 홍길동  
학점: 4.300000

# 1. 구조체의 사용

## ■ 실습 1

- 자동차 정보를 담을 수 있는 구조체를 정의해보고, 활용해보자
- 자동차 정보를 scanf를 통해 받아오고, 이를 다시 출력한다.
- 자동차 정보 멤버
  - 차종: str
  - 연식: int
  - 차량번호: str
  - 제조사: str



# 1. 구조체의 사용

## ■ 구조체를 담은 구조체

### ❖ 구조체를 멤버로 가지는 구조체

```
struct date {                // 구조체 선언
    int year;
    int month;
    int day;
};
```

```
struct student {             // 구조체 선언
    int number;
    char name[10];
    struct date dob;        // 구조체 안에 구조체 포함
    double grade;
};
```

```
struct    student    s1;    // 구조체 변수 선언

s1.dob.year = 1983;          // 멤버 참조
s1.dob.month = 03;
s1.dob.day = 29;
```

# 1. 구조체의 사용

## ■ 실습 2

- 자동차 정보를 담을 수 있는 구조체를 정의해보고, 활용해보자
- 자동차 정보를 scanf를 통해 받아오고, 이를 다시 출력한다.

### ■ 자동차 멤버

- 차종: str
- 연식: int
- 차량번호: str
- 제조사: str
- 옵션: 옵션 구조체

### ■ 옵션 멤버

- 시트 가죽: str(인조가죽, 천연가죽 등)
- 전조등: str(LED, 할로겐 등)
- 구동: int(4륜, 2륜 등)

# 1. 구조체의 사용

- 구조체 typedef
  - 보통 앞 글자를 대문자화 하여 타입을 재정의 한다.

```
#include <stdio.h>

typedef int int32; // 기존 타입의 명칭을 바꿀 수 있음

// 이를 활용하여 구조체에 적용
typedef struct student {
    int number;
    char name[10];
    double grade;
} Student;

int main(void) {
    int32 a = 5;
    Student s1 = { 201921133, "홍길동", 4.2 };
    return 0;
}
```

# 1. 구조체의 사용

## ■ 구조체 배열

### ❖ 구조체 배열 초기화

```
struct student list[3] = {  
    { 1, "Park", 3.42 },  
    { 2, "Kim", 4.31 },  
    { 3, "Lee", 2.98 }  
};
```

### ❖ 구조체 배열의 요소 개수 자동 계산

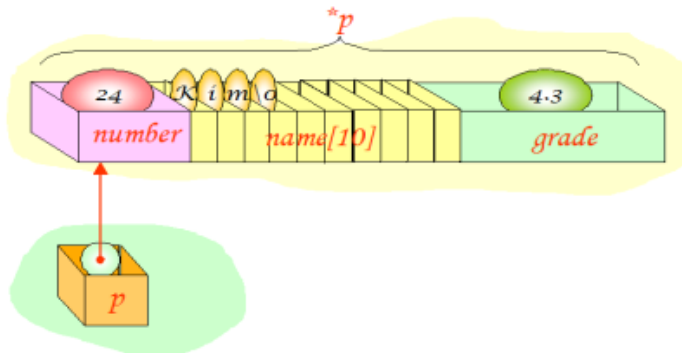
- 구조체 배열에서 요소의 개수를 자동으로 알아내려면?
- 전체 배열의 총바이트 수를 개별 요소의 바이트 수로 나누면 됨
  - ✓  $n = \text{sizeof}(\text{list}) / \text{sizeof}(\text{list}[0]);$
  - ✓ 또는  $n = \text{sizeof}(\text{list}) / \text{sizeof}(\text{struct student});$

# 1. 구조체의 사용

## ■ 구조체에 대한 포인터

### ❖ 구조체를 가리키는 포인터

```
struct student s = { 24, "Kim", 4.3 };  
struct student *p;  
  
p = &s;  
  
printf("학번=%d 이름=%s 학점=%f \n", s.number, s.name, s.grade);  
printf("학번=%d 이름=%s 학점=%f \n", (*p).number, (*p).name, (*p).grade);
```



# 1. 구조체의 사용

## ■ 구조체에 대한 포인터

### ❖ 구조체를 가리키는 포인터

- -> 연산자 (간접 멤버 연산자(indirect membership operator))
- 구조체 포인터로 구조체 멤버를 참조할 때 사용

```
struct student s = { 24, "Kim", 4.3 };  
struct student *p;  
  
p = &s;  
  
printf("학번=%d 이름=%s 학점=%f \n", p->number, p->name, p->grade);
```

p가 가리키는 구조체 변수

**(\*p).number**

p가 가리키는 구조체 변수의 멤버 number

**p->number**

p가 가리키는 구조체 변수의 멤버 number

# 1. 구조체의 사용

- 포인터를 멤버로 가지는 구조체

❖ 예제: 구조체 포인터를 멤버로 가지는 구조체

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

```
struct student {  
    int number;  
    char name[20];  
    double grade;  
    struct date *dob;  
};
```

# 1. 구조체의 사용

## ■ 실습 3

- 학생 정보를 담는 구조체가 있다
- 학생 정보 = (학번, 이름, 학점)
- 학번과 이름 학점을 간략히 하드코딩하여 학생 배열을 생성한다. (길이 5)
- 학생 배열에 담긴 내용을 출력하라.
- 학생 배열을 순회할 때, `student[i]` 연산을 사용하지 않고
- `student++`로 포인터를 옮기고, 구조체 멤버 접근에 대해서는 '`->`' 연산자를 활용한다.



# 1. 구조체의 사용

## ■ 실습 4

- 학생 정보를 담는 구조체와 날짜를 담는 구조체가 있다.
  - 학생정보 = (학번, 이름, 학점, 졸업일)
  - 날짜 = (년, 월, 일)
  - 졸업일을 날짜 구조체를 활용하여 저장한다.
- 
- 학생 구조체 변수가 3개 있다고 가정한다.
  - 한 학생의 졸업일을 변경하면, 나머지 학생의 졸업일도 동시에 변경될 수 있도록 개발한다.

# 1. 구조체의 사용

## ■ 구조체를 함수에 전달

### ■ 구조체는 call by value!

#### ❖ 구조체를 함수의 인수로 넘기는 방법

- 구조체의 복사본이 함수로 전달
- 만약 구조체의 크기가 크면 그만큼 시간과 메모리가 소요됨

구조체의 경우, 복사된다.

```
int equal(struct student s1, struct student s2)
{
    if( s1.number == s2.number )
        return 1;
    else
        return 0;
}
```

```
int main(void)
{
    struct student a = { 1, "hong", 3.8 };
    struct student b = { 2, "kim", 4.0 };
    if( equal(a, b) == 1 ){
        printf("같은 학생 \n");
    }
    else {
        printf("다른 학생 \n");
    }
}
```

# 1. 구조체의 사용

## ■ 실습 5

- 학생 정보(실습4 활용) 구조체를 함수로 넘기려고 한다.
- 학생 정보 구조체의 크기가 만약 크다면, 함수 호출에 시간이 오래걸린다.
- 학생 정보 구조체의 크기가 크다고 가정하고, 이 호출 시간을 줄이는 방법을 고안해보자.
- 호출해야 하는 함수
  - `void print_advice(학생 정보): 학생 성적에 따른 평가를 출력하는 함수`
    - If 학점 > 4: print "(이름)학생은 (학점)으로 매우 우수한 성적임"
    - If 학점 > 3: print "(이름)학생은 (학점)으로 우수한 성적임"
    - If 학점 > 2: print "(이름)학생은 (학점)으로 저조한 성적임"
    - else: print "(이름)학생은 (학점)으로 불량한 성적임 "

## 2. 공용체의 사용

### ■ 공용체

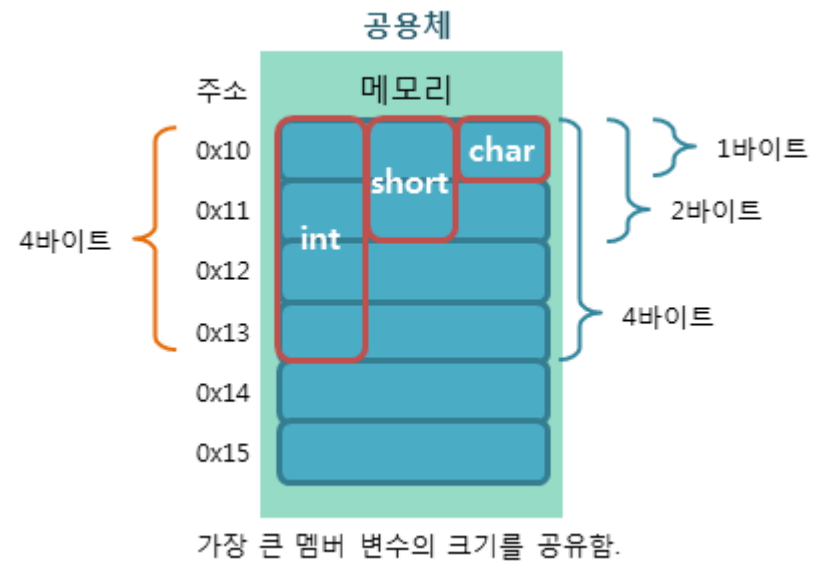
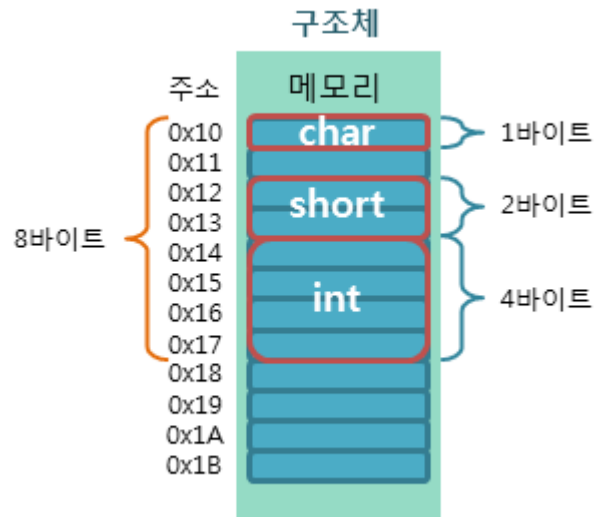
#### ❖ 공용체(union)

- 같은 메모리 영역을 여러 개의 변수가 공유
- 공용체를 선언하고 사용하는 방법은 구조체와 아주 비슷함
- 공용체 멤버 접근은 dot(.) 연산자 사용

```
union example {  
    char c;           // 같은 공간 공유  
    int i;            // 같은 공간 공유  
};  
  
union example v;  
  
union example v = { 'A' };
```

## 2. 공용체의 사용

### ■ 공용체



## 2. 공용체의 사용

### ■ 실습 6

- 공용체를 사용하여 hex로 된 ip를 전달받고, 이를 4자리의 ip로 변환해보자.
- 통신 프로그램에서는 공용체의 특성을 자주 활용한다.
- 우리도 이를 활용해보자.
- 공용체 멤버 중 unsigned int ip\_hex변수에 ip 주소가 hex 타입으로 담긴다.
- 공용체 멤버로 구조체를 선언하여, 구조체의 멤버로 각 클래스별 (‘.’으로 구분된 숫자들)로 ip 주소를 받아올 수 있어야 한다.
- 코드 예시
  - `union ip {`
  - `unsigned int ip_hex;`
  - `struct {`
  - `...`
  - `} ip_segment;`
  - `}`

## 2. 공용체의 사용

---

- 실습 6

- 입력값, 출력값

- 0x1805a5c0    192.168.5.24
    - 0x40e00457    87.4.224.64