

고급C프로그래밍

08 구조체(1/2)

(1) 구조체

변수, 배열, 구조체

- **데이터형 변수**

- ✓ 동일한 데이터 형식을 가지는 단일한 값

- **배열**

- ✓ 동일한 데이터 형식을 가지는 복수의 값

- **구조체**

- ✓ 복수의 데이터 형식을 가지는 복수의 값

- **클래스**

- ✓ 복수의 데이터 형식 값과 이를 처리하기 위한 함수로 구성

자료 구조

- **데이터를 구성하고 저장하기 위한 방식**

- ✓ 프로그램에서 데이터를 효율적으로 다루기 위한 데이터 저장 구조
- ✓ C언어에서 가장 대표적인 자료 구조 : 배열
- ✓ 배열이라는 자료 구조의 형식은 사전에 정의되어 있음

- **구조체란?**

- ✓ 처음으로 배우는 customizing 자료 구조
- ✓ 내가 원하는 조합으로 데이터를 저장, 관리하기 위한 구조를 설계

- **왜 customizing 자료 구조가 필요할까?**

- ✓ SW란 컴퓨터를 이용해서 현실의 일을 대신 처리 하기 위한 것
- ✓ 현실을 구성하는 다양한 요소들을 프로그램 안에서 묘사하기 위함

구조체 개념 (1/3)



그림 13-1 구조체와 비빔밥의 개념 비교

- **비빔밥** : 별도의 재료들을 하나의 그릇에 담고 비빔밥이라는 이름으로 정의
- **구조체** : 서로 다른 변수의 형태를 하나의 블록으로 묶음
 - ✓ 비빔밥의 당근 → ‘비빔밥 안에 있는 당근’이라고 부를 수 있음
 - ✓ 구조체 안의 변수 → ‘구조체 안에 있는 변수’라고 표현
- 이를 코드 형태로 표현하면 ‘구조체 이름.변수 이름’

구조체 개념 (2/3)

- 구조체 만들기

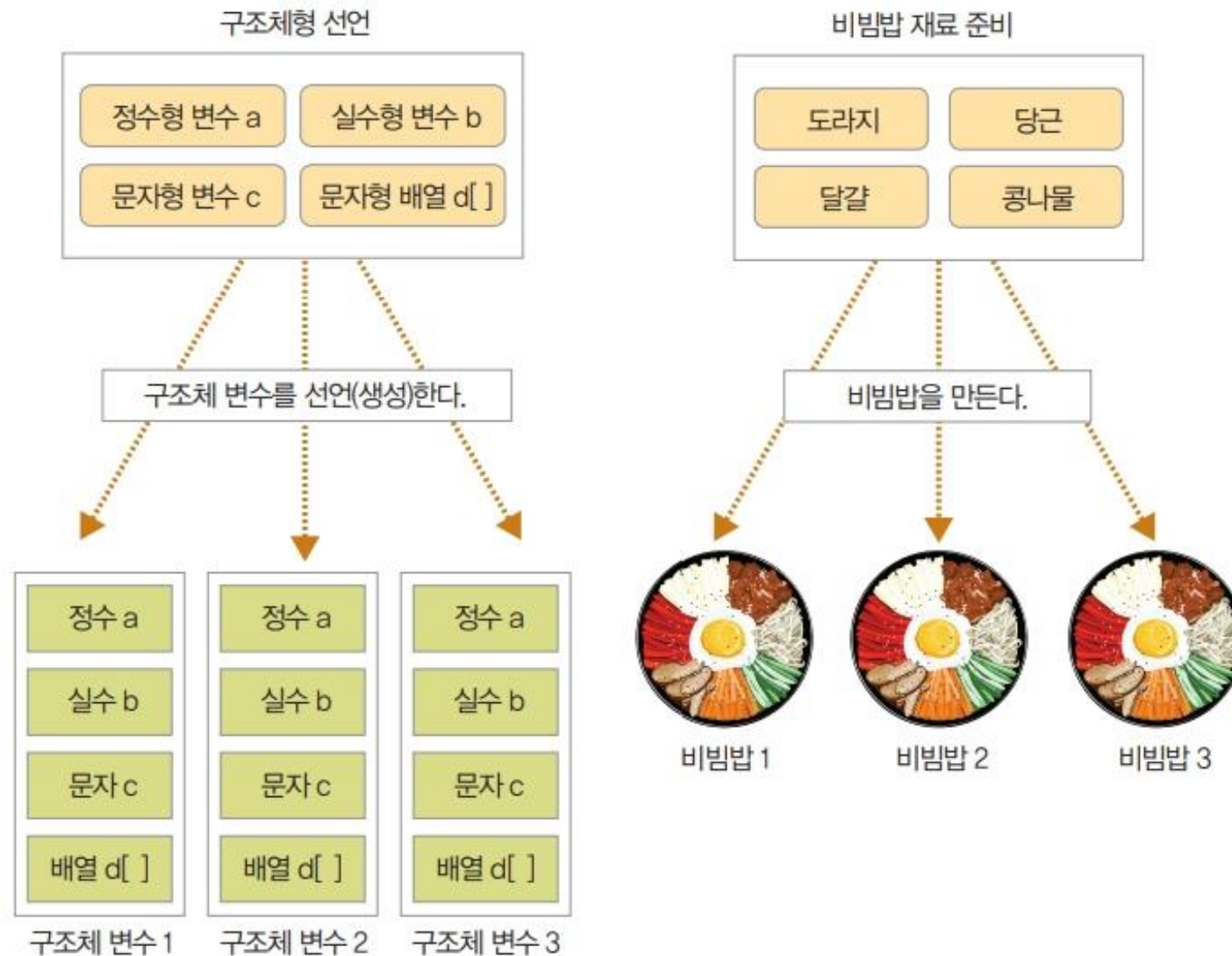


그림 13-2 구조체 변수와 비빔밥을 만드는 방법

구조체 개념 (3/3)

- 구조체 표현

① 비빔밥 만들기

```
비빔밥 재료{  
    도라지 dora;  
    당근 dang;  
    계란 egg;  
    콩나물 kong;  
};  
비빔밥 재료 비빔밥1;  
비빔밥 재료 비빔밥2;  
비빔밥 재료 비빔밥3;
```



② 구조체로 표현하기

```
struct bibim {  
    int a;  
    char b;  
    float c;  
    char d[5];  
};  
struct bibim b1;  
struct bibim b2;  
struct bibim b3;
```

① 비빔밥 재료를 준비해서 이 재료들로 비빔밥 세 그릇을 만들

② bibim 이라는 구조체형을 선언한 후 구조체 변수 세 개를 생성

구조체 문법 (1/4)

- **멤버 변수 : 구조체 안에서 정의된 변수**

- ✓ 구조체 내에서 일반적인 변수 선언과 동일한 방법으로 선언

```
struct 구조체형_이름 {  
    데이터_형식  멤버_변수 1;  
    데이터_형식  멤버_변수 2;  
    ...  
};  
  
struct 구조체형_이름 구조체_변수;
```

여기서 잠깐 구조체형과 구조체 변수

- ✓ 붕어빵(구조체 변수)을 만들고 싶으면 먼저 붕어빵 기계(구조체형)를 만들어야 함
- ✓ 붕어빵 기계(구조체형)는 한 번 만들어놓으면 붕어빵(구조체 변수)을 여러 개 찍어낼 수 있음

구조체 문법 (2/4)

- 구조체 사용 예

기본 13-1 구조체 사용 예

13-1.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 #include <string.h>
04 void main( )
05 {
06     struct bibim {
07         int a;
08         float b;
09         char c;
10         char d[5];
11     };
```

—— 구조체형 bibim을 선언한다(아직 저장 공간이 없다).

구조체 문법 (3/4)

• 구조체 사용 예 (계속)

```
12
13 struct bibim b1;      ——— 구조체 변수 b1을 선언한다(실제 저장 공간을 확보한다).
14
15 b1.a = 10;            ——— 구조체 변수의 멤버 변수에 값을 대입한다.
16 b1.b = 1.1f;
17 b1.c = 'A';
18 strcpy(b1.d, "ABCD");
19
20 printf(" b1.a ==> %d \n", b1.a);    ——— 구조체 변수의 멤버 변수값을 출력한다.
21 printf(" b1.b ==> %f \n", b1.b);
22 printf(" b1.c ==> %c \n", b1.c);
23 printf(" b1.d ==> %s \n", b1.d);
24 }
```

구조체 문법 (4/4)

• 구조체 형식

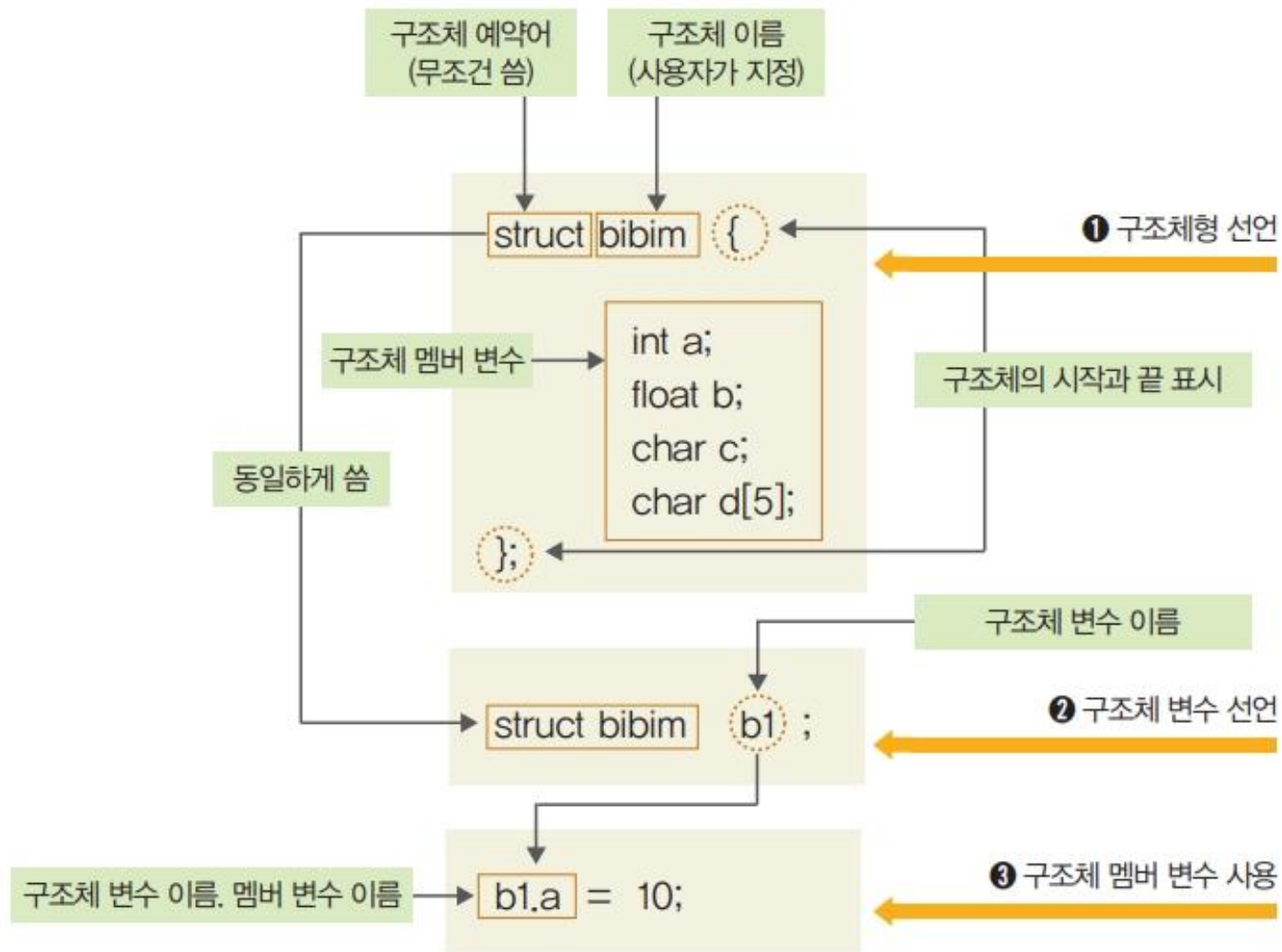


그림 13-3 구조체의 문법 구조

구조체 활용 (1/13)

• 학생의 과목별 점수와 평균 점수 관리 프로그램 작성

✓ 구조체를 사용하지 않는 기본적인 경우

기본 13-2 구조체를 사용하지 않은 예

13-2.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     char name[10];
06     int kor;
07     int eng;
08     float avg;
09
10     printf("이름 : ");
11     scanf("%s", name, 9);
12
13     printf("국어 점수 : ");
14     scanf("%d", &kor);
15
```

—— 학생 이름, 국어 점수, 영어 점수,
평균 점수 변수를 선언한다.

—— 학생 이름을 입력한다. 최대 9자를
입력할 수 있다.

—— 국어 점수를 입력한다.

구조체 활용 (2/13)

- 학생의 과목별 점수와 평균 점수 관리 프로그램 작성

✓ 구조체를 사용하지 않는 기본적인 경우

```
16  printf("영어 점수 : ");
17  scanf("%d", &eng);          ----- 영어 점수를 입력한다.
18
19  avg = (kor + eng) / 2.0f;    ----- 평균 점수를 계산한다.
20
21  printf("\n");
22  printf("학생 이름 ==> %s\n", name); ----- 학생 이름, 국어 점수, 영어 점수,
23  printf("국어 점수 ==> %d\n", kor);      평균 점수를 출력한다.
24  printf("영어 점수 ==> %d\n", eng);
25  printf("평균 점수 ==> %5.1f\n", avg);
26 }
```

실행 결과

이름 : James
국어 점수 : 87
영어 점수 : 72

학생 이름 ==> James
국어 점수 ==> 87
영어 점수 ==> 72
평균 점수 ==> 79.5

✓ 학생을 구성하는 요소들인 이름, 과목점수, 평균점수에 대해 각각 관리

➔ 비효율적이며 불편함

구조체 활용 (3/13)

- 학생의 과목별 점수와 평균 점수 관리 프로그램 작성

- ✓ 구조체를 사용하여 수정

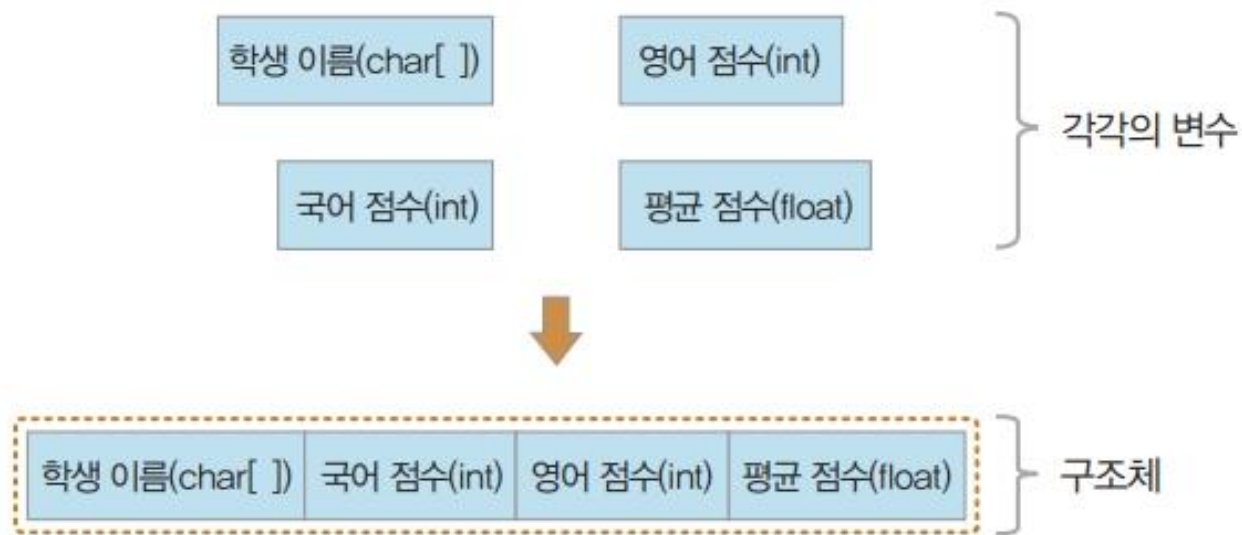


그림 13-4 구조체를 사용한 변수 선언

구조체 활용 (4/13)

• 학생의 과목별 점수와 평균 점수 관리 프로그램 작성

✓ 구조체를 사용하여 수정

기본 13-3 구조체를 사용하여 변경한 [기본 13-2]

13-3.c

```
01 #define _CRT_SECURE_NO_WARNINGS
```

```
02 #include <stdio.h>
```

```
03 void main( )
```

```
04 {
```

```
05     struct student {
```

```
06         char name[10];
```

```
07         int kor;
```

```
08         int eng;
```

```
09         float avg;
```

```
10     };
```

```
11
```

```
12     struct student s;
```

```
13
```

```
14     printf("이름 : ");
```

```
15     scanf("%s", s.name, 9);
```

```
16
```

----- student 구조체형을 선언한 후 멤버 변수를 선언한다(학생 이름, 국어 점수, 영어 점수, 평균 점수).

----- 구조체 변수 s를 선언한다.

----- 학생 이름을 입력한다.

구조체 활용 (5/13)

• 학생의 과목별 점수와 평균 점수 관리 프로그램 작성

✓ 구조체를 사용하여 수정

```
17  printf("국어 점수 : ");
18  scanf("%d", &s.kor);          ----- 국어 점수를 입력한다.
19
20  printf("영어 점수 : ");
21  scanf("%d", &s.eng);          ----- 영어 점수를 입력한다.
22
23  s.avg = (s.kor + s.eng) / 2.0f; ----- 평균 점수를 계산한다.
24
25  printf("\n--- 구조체 활용 ---\n");
26  printf("학생 이름 ==> %s\n", s.name); ----- 학생 이름, 국어 점수, 영어 점수,
27  printf("국어 점수 ==> %d\n", s.kor);      평균 점수를 출력한다.
28  printf("영어 점수 ==> %d\n", s.eng);
29  printf("평균 점수 ==> %5.1f\n", s.avg);
30 }
```

실행 결과

이름 : James
국어 점수 : 87
영어 점수 : 72

--- 구조체 활용 ---

학생 이름 ==> James
국어 점수 ==> 87
영어 점수 ==> 72
평균 점수 ==> 79.5

구조체 활용 (6/13)

• 구조체 변수의 초기화

❶ 일반 변수의 초기값 대입

```
char name[10] = "Woo";  
int kor = 90;  
int eng = 80;  
float avg;
```

→

❷ 구조체 변수의 초기값 대입

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

```
struct student s = {"Woo", 90, 80};
```

- ✓ 구조체 변수를 초기화할 때는 중괄호({ }) 안의 초기값을 콤마(,)로 분리해서 대입 → 배열 초기화와 동일
- ✓ 구조체 변수 선언 시 초기값을 일부만 정의 → 나머지는 0으로 초기화
- ✓ 초기화 하지 않을 경우 → 0으로 초기화 되지 않음

구조체 활용 (7/13)

• 구조체 변수의 다양한 선언 방법

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;
```

①은 구조체형을 만든 후 구조체 변수 s를 별도로 선언하는 방식

②는 구조체형을 선언하는 동시에 구조체 변수 s를 선언하는 방식

③은 typedef를 이용해서 struct 명령어를 이용하지 않고 변수 선언이 가능

→ struct _student를 student로 정의한다고 이해

구조체 활용 (8/13)

- “학생들”의 성적을 관리하는 것으로 확장

✓ 구조체를 사용하지 않는 경우 → 각 요소별로 배열에 저장

기본 13-4 구조체 배열을 사용하지 않은 예

13-4.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 #include <string.h>
04 void main( )
05 {
06     char name[3][10];           — 학생 이름 배열을 선언한다.
07     int kor[3];                 — 국어 점수 배열을 선언한다.
08     int eng[3];                 — 영어 점수 배열을 선언한다.
09     float avg[3];               — 평균 점수 배열을 선언한다.
10
11     int i;
12
13     strcpy(name[0], "Kim");      — 첫 번째 학생의 정보를 대입한다(학생
14     kor[0] = 90;                이름, 국어 점수, 영어 점수, 평균 점수).
15     eng[0] = 80;
16     avg[0] = (kor[0] + eng[0]) / 2.0f;
17
```

구조체 활용 (9/13)

- “학생들”의 성적을 관리하는 것으로 확장

✓ 구조체를 사용하지 않는 경우 → 배열을 이용해서 배열의 인덱스로 관리

```
18 strcpy(name[1], "Lee");
19 kor[1] = 70;
20 eng[1] = 60;
21 avg[1] = (kor[1] + eng[1]) / 2.0f;
22
23 strcpy(name[2], "Park");
24 kor[2] = 50;
25 eng[2] = 40;
26 avg[2] = (kor[2] + eng[2]) / 2.0f;
27
28 for(i=0; i < 3; i++)
29 {
30     printf("학생 이름 ==> %s\n", name[i]);
31     printf("국어 점수 ==> %d\n", kor[i]);
32     printf("영어 점수 ==> %d\n", eng[i]);
33     printf("평균 점수 ==> %5.1f\n", avg[i]);
34     printf("\n");
35 }
36 }
```

—— 두 번째 학생의 정보를 대입한다(학생 이름, 국어 점수, 영어 점수, 평균 점수).

—— 세 번째 학생의 정보를 대입한다(학생 이름, 국어 점수, 영어 점수, 평균 점수).

—— 세 번 반복하고 배열의 내용을 출력한다.

실행 결과

학생 이름 ==> Kim
국어 점수 ==> 90
영어 점수 ==> 80
평균 점수 ==> 85.0

학생 이름 ==> Lee
국어 점수 ==> 70
영어 점수 ==> 60
평균 점수 ==> 65.0

학생 이름 ==> Park
국어 점수 ==> 50
영어 점수 ==> 40
평균 점수 ==> 45.0

구조체 활용 (10/13)

- “학생들”의 성적을 관리하는 것으로 확장

✓ 학생 단위로 그룹화 해서 관리할 수 있을까? → 구조체 배열

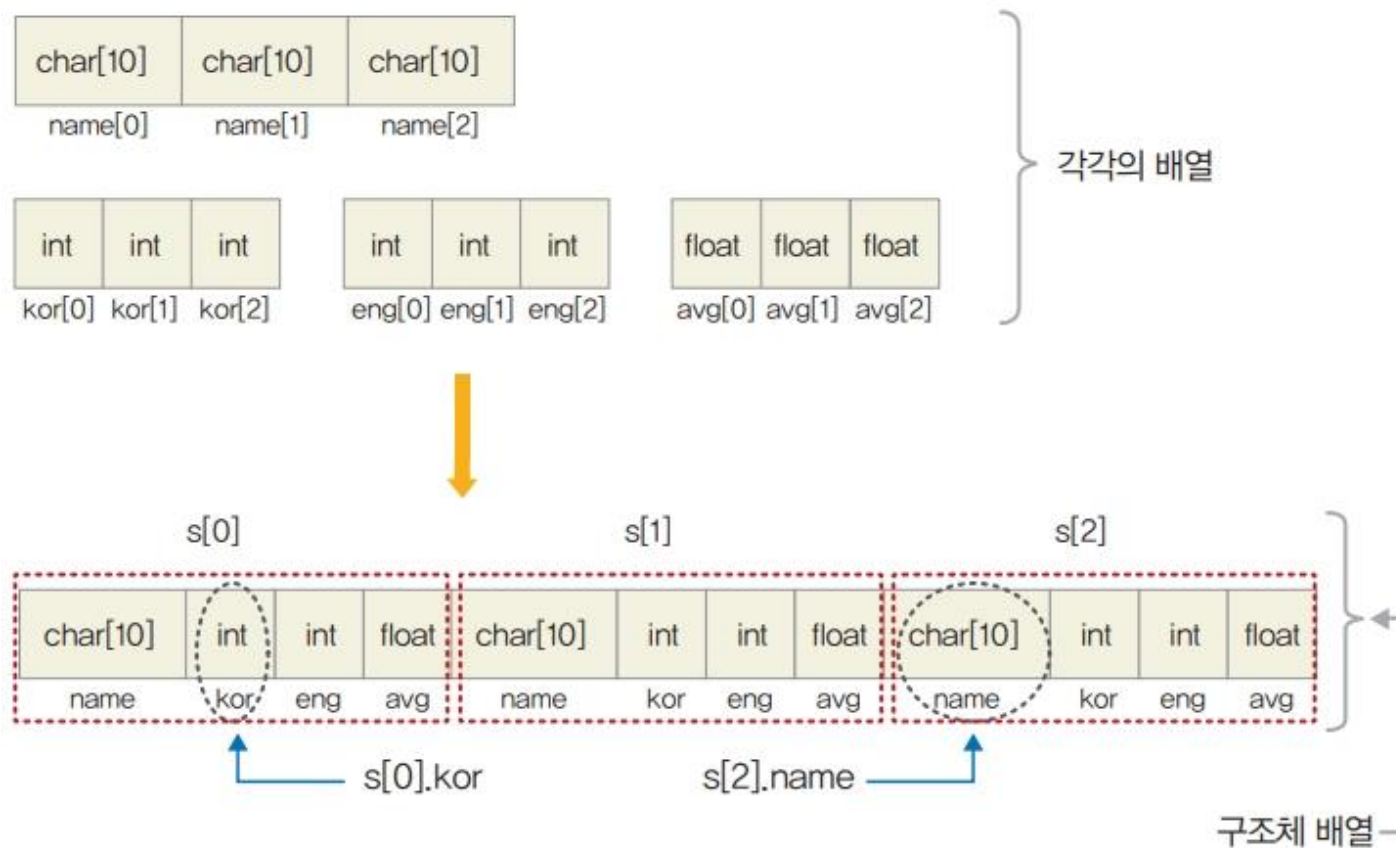


그림 13-5 구조체를 사용한 배열 선언

구조체 활용 (11/13)

- “학생들”의 성적을 관리하는 것으로 확장

- ✓ 구조체 배열을 사용

응용 13-5 구조체 배열을 사용하여 변경한 [기본 13-4]

13-5.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 #include <string.h>
04 void main( )
05 {
06     struct student {
07         char name[10];
08         int kor;
09         int eng;
10         float avg;
11     };
12
13     struct student s[3];
14
15     int i;
16
```

—— 구조체형을 선언한다.

—— 구조체 배열 s[3]을 선언한다.

구조체 활용 (12/13)

- “학생들”의 성적을 관리하는 것으로 확장

- ✓ 구조체 배열을 사용

```
17 strcpy(s[0].name, "Kim");
18 s[0].kor = 90;
19 s[0].eng = 80;
20 s[0].avg = (s[0].kor + s[0].eng) / 2.0f;
```

—— 첫 번째 학생의 정보를 대입한다
(학생 이름, 국어 점수, 영어 점수,
평균 점수).

```
21
22 strcpy( __2__, "Lee");
23 s[1].kor = 70;
24 s[1].eng = 60;
25 s[1].avg = (s[1].kor + s[1].eng) / 2.0f;
```

—— 두 번째 학생의 정보를 대입한다
(학생 이름, 국어 점수, 영어 점수,
평균 점수).

```
26
27 strcpy(s[2].name, "Park");
28 s[2].kor = 50;
29 s[2].eng = 40;
30 s[2].avg = (s[2].kor + s[2].eng) / 2.0f;
```

—— 세 번째 학생의 정보를 대입한다
(학생 이름, 국어 점수, 영어 점수,
평균 점수).

```
31
32 printf("— 구조체 배열 — \n");
33 for(i=0; i < 3; i++)
34 {
35     printf("학생 이름 ==> %s\n", s[i].name);
36     printf("국어 점수 ==> %d\n", __3__);
```

—— 세 번 반복하고 구조체 배열의
내용을 출력한다.

구조체 활용 (13/13)

- “학생들”의 성적을 관리하는 것으로 확장

- ✓ 구조체 배열을 사용

```
37     printf("영어 점수 ==> %d\n", s[i].eng);  
38     printf("평균 점수 ==> %5.1f\n", s[i].avg);  
39     printf("\n");  
40 }  
41 }
```

점명 struct student s[1].name s[i].kor

실행 결과

-- 구조체 배열 --

학생 이름 ==> Kim

국어 점수 ==> 90

영어 점수 ==> 80

평균 점수 ==> 85.0

학생 이름 ==> Lee

국어 점수 ==> 70

영어 점수 ==> 60

평균 점수 ==> 65.0

학생 이름 ==> Park

국어 점수 ==> 50

영어 점수 ==> 40

평균 점수 ==> 45.0

구조체 포인터 (1/5)

- 구조체의 주소를 저장하는 변수 : 구조체 포인터 변수

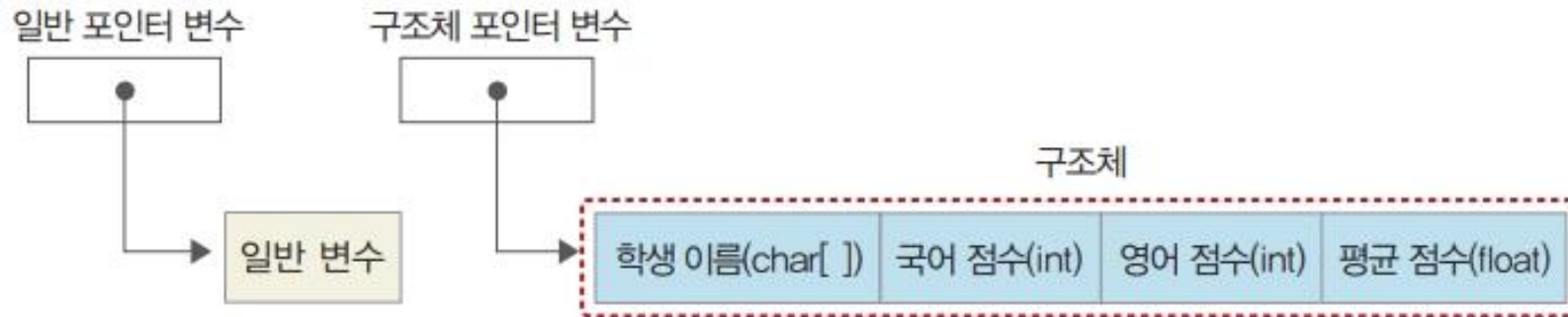


그림 13-6 일반 포인터 변수와 구조체 포인터 변수

- 일반 변수와 구조체 포인터 변수의 사용

❶ 일반 포인터 변수 사용

```
int a;  
int* p;  
p = &a;  
*p = 100;
```

❷ 구조체 포인터 변수 사용

```
struct student {  
    char name[10];  
    int kor;  
    int eng;  
    float avg;  
};  
struct student s;  
struct student *p;  
p = &s;  
p->kor = 100;
```

구조체 포인터 (2/5)

- 학생 성적 관리 프로그램을 구조체 포인터를 이용해 변경

기본 13-6 구조체 포인터 사용 예

13-6.c

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03 void main( )
04 {
05     struct student {
06         char name[10];
07         int kor;
08         int eng;
09         float avg;
10     };
11
12     struct student s;
13     struct student *p;
14
15     p = &s;
16
17     printf("이름 입력 : ");
18     scanf("%s", p->name);
19
20     printf("국어 점수 : ");
```

----- student 구조체형을 선언한 후 멤버 변수를 선언한다(학생 이름, 국어 점수, 영어 점수, 평균 점수).

----- 구조체 변수 s를 선언한다.

----- 구조체 포인터 변수 p를 선언한다.

----- 포인터 변수 p에 s의 주소를 대입한다.

----- 이름을 입력한다.

구조체 포인터 (3/5)

- 학생 성적 관리 프로그램을 구조체 포인터를 이용해 변경

```
21  scanf("%d", &p->kor);      ----- 국어 점수를 입력한다.
22
23  printf("영어 점수 : ");
24  scanf("%d", &p->eng);      ----- 영어 점수를 입력한다.
25
26  p->avg = (p->kor + p->eng) / 2.0f; ----- 평균 점수를 계산한다.
27
28  printf("\n--- 구조체 포인터 활용 ---\n");
29  printf("학생 이름 ==> %s\n", p->name); ----- 학생 이름을 출력한다.
30  printf("국어 점수 ==> %d\n", p->kor); ----- 국어 점수를 출력한다.
31  printf("영어 점수 ==> %d\n", p->eng); ----- 영어 점수를 출력한다.
32  printf("평균 점수 ==> %5.1f\n", p->avg); ----- 평균 점수를 출력한다.
33 }
```

실행 결과

이름 입력 : James
국어 점수 : 87
영어 점수 : 72

--- 구조체 포인터 활용 ---
학생 이름 ==> James
국어 점수 ==> 87
영어 점수 ==> 72
평균 점수 ==> 79.5

구조체 포인터 (4/5)

- **학생 성적 관리 프로그램을 구조체 포인터를 이용해 변경**

- ✓ 동적 메모리 할당을 이용하여 수정

```
struct student s;  
struct student *p;
```

```
p = &s
```



```
struct student *p;
```

```
p = (struct student *)malloc(sizeof(struct student));
```

구조체 포인터 (5/5)

- “학생들” 성적 관리 프로그램을 구조체 포인터를 이용해 변경
 - ✓ 동적 메모리 할당을 이용하여 수정 : 일종의 구조체 포인터 배열 사용

```
struct student s[3];
```



s에 3개 크기의 struct student 블록을 할당
→ 3개 크기의 배열과 동일

```
struct student *s;
```

```
s = (struct student *)malloc(sizeof(struct student) * 3);
```

```
.....
```

```
s[0].kor = .....
```

실습

[실습 1] 평점 계산기 업그레이드

- 7주차 [실습 2] 작성본 활용
- grade.txt로부터 과목명, 학점, 성적 데이터를 입력받아 과목 별 점수와 총 평점 계산하고, 화면에 출력
 - ✓ 각 과목별 점수 : 학점 x 성적
 - ✓ 총 평점 : 각 과목별 점수의 총합 / 학점 총합
 - ✓ 각 데이터는 Tab(\t) 1개 또는 2개로 구분되어 있음
 - ✓ 과목명과 학점, 성적, 점수는 하나의 구조체로 저장, 관리

성적	
A+	4.5
A0	4
B+	3.5
B0	3

성적	
C+	2.5
C0	2
D+	1.5
D0	1
F	0

[실습 2] 학생 이름과 나이 출력 프로그램

예제 설명 구조체 포인터와 메모리 할당 함수를 이용하여 학생의 이름과 나이를 입력받아 출력하는 프로그램이다.

실행 결과

입력할 학생 수 : 3
이름과 나이 입력 : 지효 25
이름과 나이 입력 : 정연 25
이름과 나이 입력 : 다현 23

-- 학생 명단 --

이름:지효 , 나이:25
이름:정연 , 나이:25
이름:다현 , 나이:23

[실습 3] “학생들” 성적 관리 프로그램 업그레이드

- **예제 13-6을 확장**
- **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.txt`에 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 정의하고, 사용할 것인가?**

Q & A