

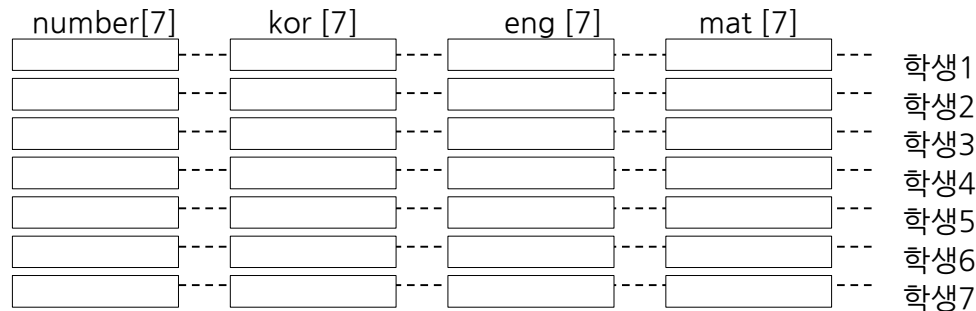
8차시 - 구조체(1)

자료의 형태

◆ 성적 처리 프로그램

- 학번(number), 국어, 영어, 수학으로 구성
- 다수의 학생이 존재

방법 1.



방법 2.

학생[0]	학생[1]	학생[2]	학생[3]	학생[4]	학생[5]	학생[6]
number	number	number	number	number	number	number
kor	kor	kor	kor	kor	kor	kor
eng	eng	eng	eng	eng	eng	eng
mat	mat	mat	mat	mat	mat	mat

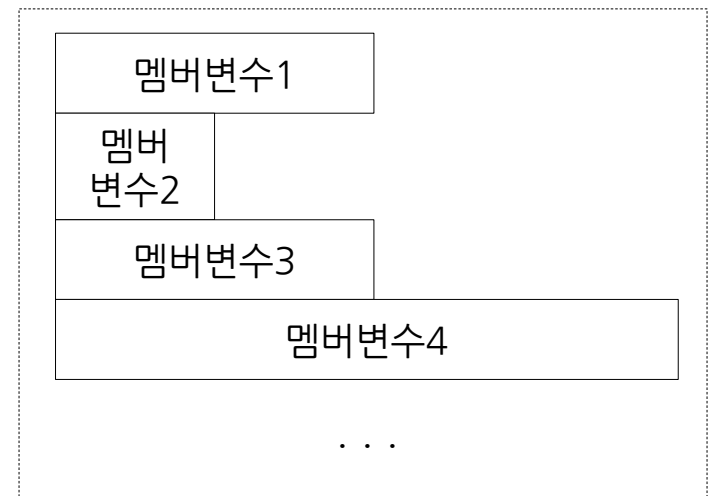
구조체

◆ 구조체

- 하나의 자료형이다.
- 다른 자료형으로 이루어진 멤버를 하나 이상 포함한다.

```
struct 구조체명 {  
    자료형1  멤버변수1;  
    자료형2  멤버변수2;  
    자료형3  멤버변수3;  
    자료형4  멤버변수4;  
    ... // 필요한 만큼 멤버를 추가  
};
```

구조체 형



구조체의 예

```
struct STUDENT {           // STUDENT라는 구조체를 새로 만든다.
    int    number;         // 번호
    int    kor;            // 국어점수
    int    eng;            // 영어점수
    int    mat;            // 수학점수 로 구성되어 있다.
};
```

◆ struct STUDENT 라는 자료형이 새로 생성되었다.

```
struct STUDENT_PERSONAL {
    int    hakbeon;
    char    name[20];
    char    address[100];
    int    age;
    float    height;
    float    weight;
};
```

구조체 변수의 선언

```
struct STUDENT {           // 구조체의 선언
    int    number;
    int    kor;
    int    eng;
    int    mat;
} student1, student2; // 변수의 선언
```

```
struct {
    int    number;
    int    kor;
    int    eng;
    int    mat;
} student1, student2;
```

구조체 변수의 선언

```
int main(void)
{
    struct STUDENT std1, std2;
    int i;
    // ...
    func ( std1 );
}

void func(struct STUDENT rstd1)
{
    struct STUDENT grade[100];
    rstd1.kor = 100;
}
```

구조체 변수의 초기화

```
struct    STUDENT    student1 = { 1000, 90, 80, 90 };
```

```
struct    STUDENT_PERSON {  
    int    number;  
    char    name[10];  
    char    sex;  
    int    height;  
}    studentvar = { 1000, "KIM", 'M', 175 };
```

구조체 배열

```
struct STUDENT students[ ] = {  
  
    { 1000, 100, 90, 90 },  
    { 1001, 90, 90, 80 },  
    { 1002, 80, 70, 90 },  
  
};
```


구조체의 사용

멤버 변수 접근 연산자

↑

student1 . kor = 10;

구조체 변수 멤버 변수

student1			
number	kor	eng	mat
	10		

◆ 다음의 의미는?

- student1.kor = 20;
- total = student1.kor + student1.eng + student1.mat;
- students[0].kor = 10;
- students[10].subject[2] = 10;
- students[3].member.subject.kor.score = 10;

◆ 멤버 변수에 값을 넣는 방법은 하나씩 채우는 방법 뿐이다.

- `student1.number = student2.number;`
- `student1.kor = student2.kor;`
- `student1.eng = student2.eng;`
- `student1.mat = student2.mat;`

◆ 구조체는 하나의 자료형이다.

- 대입문도 가능하다.
- `student1 = student2;`

```
struct people {  
    char name[10];  
    int age;  
} personA, personB;  
  
personA = personB;
```

◆ 구조체의 구성과 무관하게 대입된다. 단, 동일한 구조체에서.



`personA = personB` 하면 전체 내용을 그대로 복사한다.

◆ 구조체의 이름이 다르면 구조가 동일해도 다른 자료형이다.

– 네 개는 모두 다른 자료형이다.

```
struct TYPEA {  
    char    name[10];  
    int     age;  
} aVar;
```

```
struct {  
    char    name[10];  
    int     age;  
} ano_aVar;
```

```
struct TYPEB {  
    char    name[10];  
    int     age;  
} bVar;
```

```
struct {  
    char    name[10];  
    int     age;  
} ano_bVar;
```

– bVar = aVar; // 컴파일 오류.

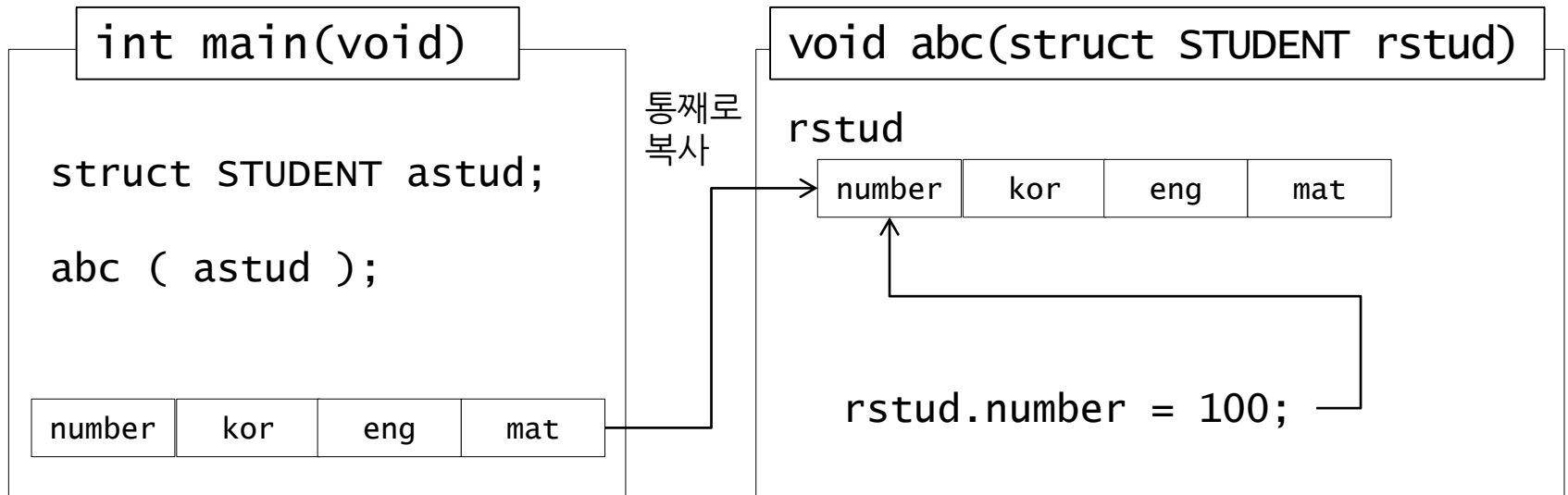
구조체의 사용

- ◆ `printf(“%d\\n”, sizeof (struct STUDENT));`
- ◆ `printf(“%d\\n”, sizeof (student1);`
- ◆ `struct STUDENT Aarray[10], Barray[10];`
- ◆ `Aarray = Barray;`
- ◆ `struct STUDENT student1, student2;`
- ◆ `if (student1 == student2)`

매개변수로서의 구조체

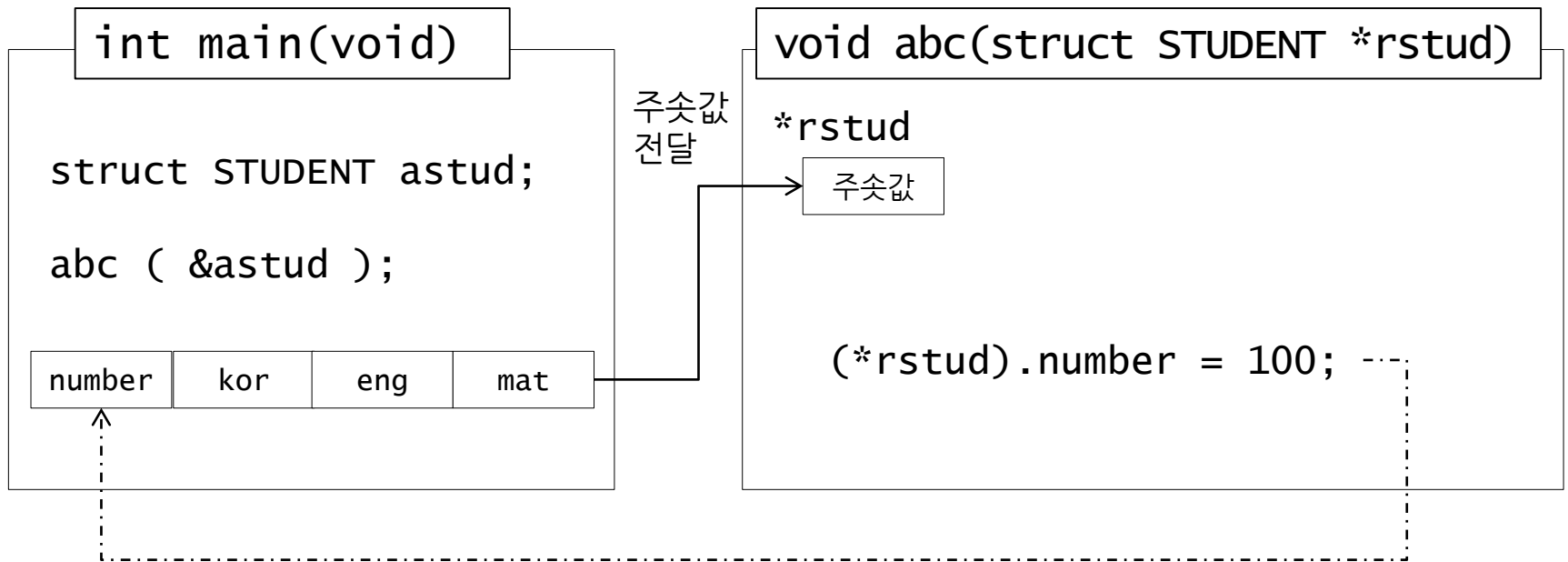
◆ 구조체는 하나의 자료형이다.

- int 형처럼 생각하라.



매개변수로써의 구조체

- ◆ 포인터로 전달할 수도 있다.



- ◆ 구조체의 크기가 크다면?

구조체와 typedef

```
typedef struct STUDENT {  
    . . .  
} StudentType;  
  
StudentType std1, std2;
```

- ◆ typedef 를 이용하여 구조체를 더 자연스럽게 만든다.

Quiz)

◆ 다음 코드의 문제는?

```
struct STUDENT {  
    int number;  
    char      *name;  
    int rank;  
};  
int main(void)  
{  
    struct STUDENT kim = {1001, "kim", 1};  
    struct STUDENT lee;  
  
    lee = kim;  
    lee.number = 1002;  
    lee.rank = 2;  
    strcpy(lee.name, "lee");  
}
```

구조체의 이름

◆ 구조체의 이름은 다른 식별자(변수명)와 구별된다.

– 구조체 이름을 변수로 쓰더라도 무관하다(좋지는 않다)

◆ 구조체 이름의 양식을 정하는 것이 좋다.

```
struct _STUDENT {  
    int    kor;  
    int    eng;  
} ;  
  
typedef struct _STUDENT Student;
```

규칙은?