

교학습내용



할 학습목표

- ♀ 11.5 구조체와 함수의 관계를 이해한다.





C프로그래밍및실습





- 1. 구조체 개요
- 2. 구조체의 정의, 선언, 사용
- 3. 구조체 배열
- 4. 구조체 포인터
- 5. 구조체와 함수
- 6. 중첩 구조체 및 자기참조 구조체
- 7. typedef 사용자 형정의

교 구조체를 함수의 인자로 사용

실인자의 값이 형식인자에 대입됨

✓ 구조체 변수라고 달라지는 건 없음

```
void print(struct student st) {
  printf("id: %d\n", st.id);
  printf("name: %s\n", st.name);
   printf("grade: %.2f\n", st.grade);
                구조체의 값 대입
int main()
   struct student st1 = {10, "Tom", 3.2};
  print(st1);
   return 0;
```

실행 결과 id: 10 name: Tom grade: 3.20 "Tom" 3.2 st 10 print의 변수 "Tom" st1 10 3.2 main의 변수



● 구조체 전체 값이 통째로 호출함수에게 전달됨

```
실행 결과
struct student init( ) {
  struct student st = \{0, "", 0\};
                                                 10, Tom, 3.20
  return st; // 구조체 변수 st의 값 반환
                                                 0,,0.00
int main() {
  struct student st1 = {10, "Tom", 3.2};
  printf("%d, %s, %.2f\n", st1.id, st1.name, st1.grade);
  st1 = init(); // init()의 반환값 대입
  printf("%d, %s, %.2f\n", st1.id, st1.name, st1.grade);
  return 0;
```



※실습하기



[예제 11.6]

두 복소수를 입력 받아, 두 복소수의 합을 출력하는 프로그램을 작성하시오.

- 구조체 complex 사용 (예제 11.3 참고)
- add_complex() 함수를 정의하여 사용할 것
 - ✓ 구조체 complex형 변수 두 개를 인자로 받아, 두 복소수의 덧셈 결과 (구조체 complex형)를 반환
- main() 함수
 - ✓ 사용자로부터 복소수의 정보를 입력 받아 구조체 변수에 저장
 - ✓ add_complex 함수를 호출하여 덧셈 결과 얻기
 - ✓ 덧셈 결과 출력

입력

1.2 2.0

-2.2 - 0.3

실행 결과

-1.01.7



구조체 <u>포인터</u> 변수를 함수 인자로 사용

실인자의 값(주소)이 형식인자에 대입됨

```
void init p(struct student *pst) {
                                                           pst
  pst->id = 0;
                                                                      실행 결과
                                                       0x3C
  pst->name[0] = '\0';
                                                  init p의 변수
                                                                     0, , 0.00
  pst->grade = 0.0;
int main() {
                   구조체의 주소 대입
  struct student st1 = {10, "Tom", 3.2};
                                                                 st1
                                             0x3C
  init_p( &st1 );
                                                      "Tom"
                                               10
                                                                 3.2
  printf("%d, %s, %.2f\n", st1.id,
                                                  main의 변수
          st1.name, st1.grade);
  return 0;
                                                init_p 함수 시작 시
                                                    메모리 그림
```



구조:

구조체 <u>주소</u>를 반환하는 함수

● 일반 변수의 주소 반환과 동일

```
struct student *next_addr(struct student *pst) {
  return pst+1;
int main() {
   struct student st[2] = \{\{10, \text{"Tom"}, 3.2\}, \}
                                     {20, "Ann", 3.5}};
   struct student *p;
   p = next_addr(st);
                                // 배열의 이름(주소) 전달
   printf("%d, %s, %.2f\n", p->id, p->name, p->grade);
   return 0;
```

실행 결과

20, Ann, 3.50



※실습하기



[예제 11.7]

두 복소수를 입력 받아, 절대값이 큰 복소수를 출력하는 프로그램을 작성하시오.

- 구조체 complex 사용 (예제 11.3 참고)
- larger_complex() 함수를 정의하여 사용할 것
 - ✓ 구조체 complex형 포인터 변수 두 개를 인자로 받아, 절대값의 제곱이 큰 복소수의 포인터를 반환
 - ✓ 참고) 복소수 a+bi의 절대값의 제곱은 a²+b²
- main() 함수
 - ✓ 사용자로부터 복소수의 정보를 입력 받아 구조체 변수에 저장
 - ✓ larger_complex 함수를 호출하여 절대값이 큰 복소수의 주소 얻기
 - ✓ 절대값이 큰 복소수 출력

입력

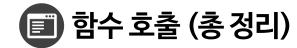
1.2 2.0

-2.2 - 0.3

실행 결과

1.2 2.0





- 함수 인자나 반환 값의 자료형에 관계없이 함수 호출 과정은 동일
- 함수 호출 시에는 <mark>전달된 값을 형식인자에 대입</mark>
 - ✓ 차이가 나는 이유는 전달된 값이
 - ✓ 정수, 문자를 나타내느냐
 - ✓ 주소를 나타내느냐에 따른
 - ✓ 부수적인 효과일 뿐…
- 함수 종료 시에는 리턴 값이 통째로 호출함수에 전달
- 결론
 - ✓ 자료형(int, 포인터, 구조체, 배열)마다 따로 외우려고 하지 말고 원리를 이해하자!!!!



C프로그래밍및실습





- 1. 구조체 개요
- 2. 구조체의 정의, 선언, 사용
- 3. 구조체 배열
- 4. 구조체 포인터
- 5. 구조체와 함수
- 6. 중첩 구조체 및 자기참조 구조체(1)
- 7. typedef 사용자 형정의



6. 중첩 구조체 및 자기참조 구조체

■ 중첩 구조체(nested structure)

- 다른 **구조체**가 구조체의 멤버로 사용
- 중첩 구조체 정의 예

```
struct address { // 구조체 address 정의 int zipcode; char *city; }; struct student5 { // 구조체 student 정의 int id; char name[8]; double grade; struct address addr; // 멤버 addr의 자료형은 struct address };
```

```
id name[8] grade addr

zipcode *city
```



6. 중첩 구조체 및 자기참조 구조체

1 2

중첩 구조체 사용 예

```
struct student5 st = {10, "Tom", 3.2, {1425, "Seoul"}};
st.addr.zipcode = 7189;  // 중첩 구조체 멤버 접근
st.addr.city = "Incheon";
printf("id: %d\n", st.id);
printf("name: %s\n", st.name);
printf("grade: %.2f\n", st.grade);
printf("zipcode: %d\n", st.addr.zipcode);
printf("city: %s\n", st.addr.city);
```

st 10 "Tom" 3.2 1425 \

"Seoul"



6. 중첩 구조체 및 자기참조 구조체

(주의) 자신과 동일한 구조체 자료형은 멤버로 사용될 수 없음

- 왜? 자기 자신을 정의하기 위해 자기 자신이 필요 (순환 오류..)
 - ✓ 아래 구조체의 메모리 그림을 그릴 수 있는가?

```
struct student6{ // 구조체 student 정의 int id; char name[8]; double grade; struct student6 roommate; // 컴파일 오류 struct student6 friends[10]; // 컴파일 오류 };
```

- ✓ 구조체 배열도 마찬가지
- ✓ 그런데, 구조체 포인터는? (다음 강의)



高い 학습 정리

- 구조체 또는 구조체 포인터를 함수에 인자로 전달하고 반환하는 과정은 일반 변수와 동일
- 함수 인자나 반환 값의 자료형에 관계없이 함수 호출 과정은 동일
- 중첩 구조체는 구조체가 멤버로 사용된 구조체
- 자신과 동일한 구조체 자료형은 멤버로 사용될 수 없음

