

### 3장 클래스

C언어 : 절차지향 언어 → C++ : 객체지향 언어.

↳ class ≡ 구조체 + 함수. (특: 접근제어, 상속)

(typedef, struct, class ... → 제작자 코드 <자료구조>  
main → 사용자 코드 [자료  
사용코드 (함수 등)  
자료의 변화 → 코드 변화 불가  
의존도 ↓ 함수를 유지보수 비용 ↓

제작 → 대상 → 사용 : 이해도 ↓  
↳ 이해도 ↑  
↳ 실수 오류 가능성 ↑

↳ 가능성 불신, 감소. ⇒ 해결책 : 생성자 클래스.  
수준 사용법 제시 필요.

C++ → 구조체 없음, only class.

Hidden Parameter : class의 주소 / this 포인터 : 객체 자체를 가리키는 포인터.

\* 멤버 선언 및 정의.

멤버 [ Data  
Function ← Method

class  
(object (객체)  
내부.

제작자 → 대상체 ← 사용자.  
실수 오남용  
접근 통제.

\* 생성자, 소멸자 ⇒ 함수, but 특수.

↳ ① 호출시점 X ② 반환 X ③ 자동 호출.

(call : 시점 or 상황을 특정)

문법에 따라 시점이 특정되어 있음.

관리의 편리성.

\* 생성자 초기화 목록.

C++ ⇒ int b(0); 초기화 시킬. 주석.

↳ 초기화 담당, 객체 생성, 소멸자에 자동 호출. \* 소멸자 : ~ class / 지역변수 소멸시점 : 소멸자 call.

★ Class "권력"으로 생성시 main 호출 전에 생성자 호출될 수 있음.

malloc → only 메모리 할당 ⇒ 생성, 소멸자 X

[ new/delete → 객체 생성 ⇒ 생성, 소멸자 O

\* 명시적 디폴트 생성자.

↳ 생성자 (Void) = default. ⇒ 선언 및 정의 둘 다.

↳ 생성자 (Void) = delete ⇒ 생성자 삭제.

\* This ⇒ 개별 인스턴스 각각에 대한 메모리 주소.

↳ 자동으로 결정.

\* 상수형 메서드 ⇒ 멤버 함수 안에서 const 선언

⇒ 멤버 변수 읽기 O, 쓰기 X

↳ this 포인터가 가리키는 대상의 상수화

\* 정적 멤버 : 선언 시 Static 선언 (전역변수, 함수) 생성 // 반드시 선언과 정의 분리.

↳ Instance 와 관련 X, this 포인터 사용 불가. 소속된 class 관리적 차원.

Instance 의 전역 <Static> ↔ <Public> 고상 전역 변수.

정적 Data 변수. → 동시성 문제 야기 → 병렬, 동시성에 각별함.

Multi Threading → thread local storage