

# Class

EUnS

December 31, 2019

## ① OOP

## ② class

# OOP란?

- 객체 지향 프로그래밍(Object oriented programming)
- 사실 본인도 잘 모름
- 어떤 물체를 클래스 설정하고 애를 메소드(멤버 함수)로 가지고노는것.
- 큰 단위의 프로젝트의 협업을 위한것(큰 코드를 다룰때) : 추상화
- 그럼에도 클래스 내부를 이해하는것은 중요하다.

[참고1](#) [참고2](#)

## class

```
1  class name
2  {
3  public:
4      name();
5      ~name();
6      void function1();
7      void function2();
8      void function3();
9  private:
10     int memberVariable1;
11     char memberVariable2;
12     void function4();
13 };
14 void name::function1() { }
15
```

# 용어 정리

- 멤버 함수(member variable)
- 메소드(method) = 멤버 함수(member function)
- public : 외부에서 자유롭게 사용할수있는 것들
- private : class내 public에서 접근가능
- 객체 = 오브젝트(object) = 인스턴스(instance)
- 캡슐화(encapsulation) : 대충 숨긴다는 뜻
- 정보 은닉 : 대충 숨긴다는 뜻
- protect : 상속 받은 자식에서도 접근가능
- 다형성 : 상속을 통한 다양함을 나타낸다는 OOP 특징

# 생성자 소멸자

- 초기화 : 멤버 이니셜라이저
- 반환값 X

```
1      name::name(int a, char b): memberVariable1(a),  
    memberVariable2(b)  
2      {  
3          ;  
4      }  
5      name::~~name()  
6      {  
7          delete [] something;  
8      }  
9
```

## class 메모리 구조

- ① 변수 선언순으로 올라감.
- ② 함수 .text 영역에
- ③ static 변수는 .BSS 영역

## C vs Cpp struct, class와 차이

- ① Cpp에서는 struct에 메소드 선언이 가능하다.
- ② C방식으로 struct 안붙이고 뒤에 이름만 사용가능
- ③ 기본접근자가 public임 class는 private
- ④ struct는 C 스타일로만 사용



# 과제

과제 1 auto, using namespace std; 쓰지말것