

# <데이터 구조> : 5공 604호.

Date : 2020-03-18

과목 순수 이론 과목. → 과제는 무조건 코딩. 전공 필수 이므로, 어과목에서 프로그래밍 꼭 배워야 함.  
상향된 프로그래밍 스킬이 필수여야 함.

언어 C++ 기반으로 강의 실시. (O.O.P), (C++론 개념 꼭 공부함).

효율적 코딩, 알고리즘과. 잘 알려진 알고리즘으로 각론은 코딩 구현, 성능 향상이 얼마나 되게?

data structures and algorithms. fourth edition.

공간 (30%) / 기성 (30%)

속도 / 연습 (30%) 발표 / 평가 (10%).

C++ 기본 encapsulation, inheritance, pointer, overloading function, polymorphism, standard template library.

2주 내용

알고리즘 + 자료구조. (complexity analysis)

Data structure, Algorithm.

OOP object-oriented programming.

data + function ⇒ object. (C++) \* object = instance of class  
(가공) class의 인스턴스.

class = struct.

encapsulation. 캡슐화로 데이터 보호, \* 접근자 : 외부 접근 권한 설정을 데이터 보호.

public 공개, 상수적

class. #include <iostream>

using namespace std;

class C  
{

public: // class 외부에서 접근 가능한 것

int a; void fun1 (int input)

int b; C = input;

d = input

protected: // class 외부에서 읽기, 쓰기 모두 불가!

int c;

상속에서 개념 차이.

private: // class 외부에서 읽기, 쓰기 모두 불가!

int d;

};

int main ()

{

(class)

C my\_object; // class (기본) object 변수 선언. // like someone.

my\_object.a = 10; // class에서 int a = 10.

C my\_object2;

my\_object2.a = 20; //

class 함수는 대부분 public 에서 선언됨.

data member는 protected, private에 선언.

생성과 함수.

class person. person () 이 동적래 있음.  $\Rightarrow$  person::person () 이다.

public :

void set\_age (int a); void set\_height (int a);

private :

int m\_age; // 개인정보에 보통 prefix 설정

int m\_height;

};

void person :: set\_age (int a) // class 밖에서 함수 선언.

{

m\_age = a;

}

int main () {

person pl, p2;

p1.set\_age = ?

p1.set\_height = ?

}

class template.

template <class type>

타입 식별자 (변경가능).

class numbers

(오류의 특징)

1개의 기능으로 여러개의 함수 재사용가능.

{

public:

type a;

};

int main () {

numbers <int> int\_class;

numbers <double> double\_class;

int\_class.a = ?

double\_class.a = ?

}

inheritance 상속.

원본 클래스에서 받고 싶은 것만 받고, 수정하는 것.

Date : 2020-03-19