**<참조와 const>**

const int &ref = 3+4;

//int &ref의 경우 Rvalue를 Literal Initialization이 불가능 하지만, const의 경우 가능.

const int ref = 3+4;

언뜻보면, const int ref와 const int& ref는 차이점이 없어 보인다.

void doSth(const int ref){…} //변수가 복사된다.

void doSth(const int& ref){…} //변수복사가 발생하지 않아 메모리 면에서 효율적이다.

//Person 구조체

int age;

double weight;

// In main

Person P;

P.age = 5;

P.weight = 30;

// 일반적인 구조체 운용

Person &ref = P;

ref.age = 10;

ref.weight = 50;

// 구조체형 참조변수를 사용

Person \*ptr = &P;

ptr->age = 20;

ptr->weight = 100;

// 구조체 포인터 사용 참조 연산자 '->'

(\*ptr).age = 30;

(\*ptr).weight = 50;

// 구조체 포인터 사용 (\*ptr).멤버이름

cout << (uintptr\_t)&P << endl;

cout << (uintptr\_t)&ref << endl;

cout << (uintptr\_t)ptr << endl;

// result

11074116 // &p

11074116 // &ref

11074116 // ptr

※동적할당배열에서는 for-each문 사용할 수 없다.

-> std::vector의 등장배경