

# Day-28

## Initialization Structure Initialization, Dynamic Memory Initialization

### Structure Initialization

- C++03

```
struct S{
    unsigned x;
    unsigned y;
};

S s1;           // 기본 초기화 x,y는 정의되지 않은 값
S s2 = {};      // 복사 목록 초기화 x,y 0/기본 초기화
S s3 = {1, 2};  // 복사 목록 초기화 x=1, y=2
S s4 = {1};     // 복사 목록 초기화 x=1, y 0/기본 초기화
// S s5(3, 5);  // 컴파일 에러

S f() {
    S s6 = {1, 2};
    return s6;
}
```

- C++11

```
struct S{
    unsigned x;
    unsigned y;
    void* ptr;
};
```

```

S s1{}; // 직접 목록(또는 값) 초기화
        // x, y, ptr 0/기본 초기화
S s2{1, 2}; // 직접 목록(또는 값) 초기화
        // x=1, y=2, ptr 0/기본 초기화
// S s3{1, -2}; // 컴파일 에러

S f() { return {3, 2}; }

```

## Structure Initialization - Brace or Equal Initialization

- 중괄호 또는 동등 초기화라고도 하는 NSDMI(비 정적 데이터 멤버 초기화)

```

struct S{
    unsigned x = 3; // 동등 초기화
    unsigned y = 2; // 동등 초기화
};
struct S1{
    unsigned x{3}; // 중괄호 초기화
};

S s1; // 기본 생성자 호출 (x=3, y=2)
S s2{}; // 기본 생성자 호출 (x=3, y=2)
S s3{1, 4} // set x=1, y=4

```

## Structure Initialization - Designated Initializer List

- C++20에 지정된 초기화 목록 도입

```

struct A{
    int x, y, z;
};

A a1{1, 2, 3};
A a2{.x = -1, .y = 2, .z = 3} // 지정된 초기화 목록

```

- 지정된 초기화 목록은 코드 가독성 향상에 매우 유용할 수 있음

```

void f1(bool a, bool b, bool c, bool d, bool e){}
// 동일한 데이터 타입의 긴 list -> 오류 발생 가능성 높음

struct B{
    bool a, b, c, d, e;
};
// f2(B b)
f2({.a = true, .c = true}); // b, d, e = false

```

## Structure Binding

- C++17의 구조 바인딩 선언은 지정된 이름을 이니셜라이저의 요소에 바인딩함

```

struct A{
    int x= 1;
    int y= 2;
}a;

A f() { return A{4, 5}; }
// Case1 : struct
auto [x1, y1] = a;    // x1=1, y1=2
auto [x2, y2] = f(); // x2=4, y2=5
// Case2 : raw arrays
int b[2] = {1,2};
auto [x3, y3] = b;    // x3=1, y3=2
// Case3 : tuples
auto [x4, y4] = std::tuple<float, int>{3.0f, 2};

```

## 동적 메모리 초기화

- C++03

```

int* a1 = new int;           // undefined
int* a2 = new int();         // 0-초기화
int* a3 = new int(4);        // 4 단일 값 할당
int* a4 = new int[4];        // 정의되지 않은 값으로 4개 요소 할당

```

```
int* a5 = new int[4]();    // 4개 요소 0-초기화  
// int* a6 = new int[4](3); // 유효하지 않음
```

- C++11

```
int* b1 = new int[4]{};    // 4개 요소 0-초기화  
int* b2 = new int[4]{1, 2}; // 첫번째 1, 두번째 2, 나머지 0-초기화
```