

Basic_Concepts_3 - Non-allocating Placement Allocation | Non-throwing Allocation

Non-allocating Placement

(ptr) type 로 이미 할당된 오브젝트의 메모리 위치를 명시할 수 있다.

```
// 스택 메모리
char buffer[8];
int* x = new (buffer)int;
short* y = new (x + 1) short[2];
// x, y는 해제할 필요가 없다.

// 힙 메모리
unsigned* buffer2 = new unsigned[2];
double* z = new (buffer2) double;
delete [] buffer2; // ok
// delete [] z ok 가능은 하지만, 나쁜 습관임.
```

non-trivial object의 배치할당은 오브젝트가 스코프 밖으로 벗어난것을 런타임중에 감지할 수 없기 때문에 명시적으로 소멸자를 호출해야한다.

```
struct A {
    public ~A() { cout << "distructor" << endl;
}

char buffer[10];
auto x = new (buffer) A();
// delete x; // error! x는 힙메모리에 올라온 변수가 아니므로 에러 발생
x->~A(); // 소멸자 호출
```

`new` 연산자는 `std::nothrow` 오브젝트의 non-throwing 할당을 허용한다. 만약 메모리 할당에 실패하면, `std::bad_alloc` 예외를 던지는 대신 `NULL` 포인터를 반환한다.

```
int* array = new (std::nothrow) int[very_large_size]; // NULL
```

`std::nothrow`가 예외를 던지지 못한다는 의미는 아니다.