C프로그래밍 실습

**#11 Dynamic Memory Allocation**

* Storage Class Review

+ 변수는 변수의 범위와 생존 시간으로 범주화 된다.

- local (automatic) variable: 지역변수

- global variable: 전역변수

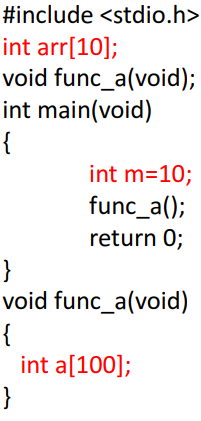
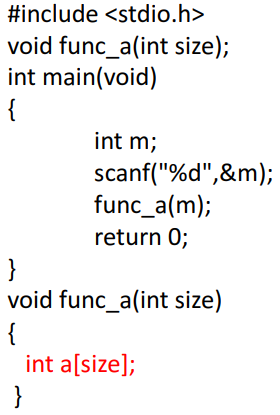
- static local variable: 정적 지역 변수

+ 동적 메모리는 또 다른 유형의 storage class이지만 변수가 아니라 메모리다

- 동적 메모리 함수에 의한 문장으로 실행 시간 (on-demand)을 할당 / 해제한다.

- 할당 단위는 byte이다.

* Data Variable size



1. 변수의 크기가 컴파일 시간에 프로그램 문으로 결정된다.
2. 다음과 같은 소스코드를 컴파일 하면 실행이 안된다.

int[size]: 변수의 크기는 프로그램을 작성할 때 결정해야 한다. 프로그램 문은 컴파일 될 때 코드화 된다.

* Dynamic Memory

+ Static memory (local or global variable)

변수에 대한 메모리 할당 시간과 크기는 컴파일 타임에 결정된다.

(프로그래머가 프로그램 문을 작성하는 동안)

+ Dynamic Memory

변수에 대한 메모리 할당 시간과 크기는 런타임에 결정된다.

(프로그램이 실행되는 동안)

|  |  |  |
| --- | --- | --- |
|  | Static memory | Dynamic memory |
| Allocation | 변수가 필요할 때 컴파일러에 의해 할당 | 런타임에 명시적으로 프로그래머에 의해 할당 |
| De-allocation | 컴파일러에 의해 자동 반환 | 런타임에 명시적으로 프로그래머가 반환 |
| Scope | 지역변수: 프로그램 블록 내  전역 변수: 소스코드 내 | 메모리 공간에 대한 포인터를 사용할 수 있는 프로그램 문 |
| Life-time | 컴파일러에 의해 생성 및 삭제 됨 | 프로그래머에 의해 생성 및 삭제 됨 |

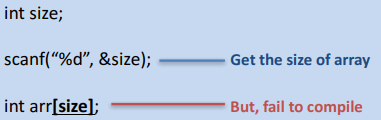
+ 동적 메모리의 범위

할당 된 동적 메모리가 활성 상태일 때 포인터를 사용할 수 있는 모든 프로그램 문

(같은 프로그램 문이라도 동적 메모리의 수명에 따라서 사용여부가 다르다)

+ 동적 메모리의 수명

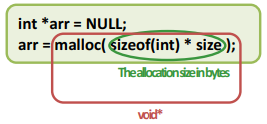
프로그래머가 동적 메모리 공간을 할당 해제 할 때까지

* Necessities of Dynamic Memory

메모리 공간이 낭비되거나 최대 배열 크기가 컴퓨터에 맞지 않는다.

배열의 크기는 상수가 된다.

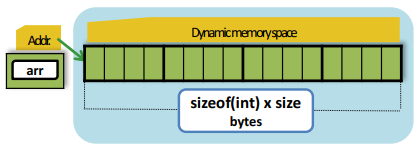
* Dynamic Memory Allocation

malloc()을 이용한다.

#include <stdlib.h>파일을 써야한다.

* void \*malloc(size\_t size);

사이즈는 byte단위로 할당한다

할당된 메모리에 대한 포인터를 반환한다.

+ 동적으로 할당된 메모리에 대한 포인터를 저장하는 포인터 변수를 선언한다.

* int \*arr = NULL;

+ 크기가 sizeof(int) \*size 인 메모리 공간을 동적으로 할당하고 arr의 공간에 포인터를 저장한다.

* arr=malloc(sizeof(int)\*size);

+ 동적으로 할당 된 메모리 공간 (“size” bytes)의 포인터(시작 메모리 주소)를 반환.

+ 요청한 크기를 할당하지 못하면 NULL을 반환한다.

컴퓨터에 메모리가 가득 차 있거나 너무 많은 메모리를 할당하려고 할 경우

* Dynamic Memory De-allocation

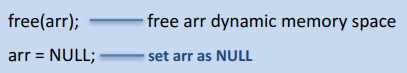
+ 할당 된 메모리가 프로그램에서 더 이상 필요 없다면 free()에 의해 메모리가 반환 (de-allocation) 될 수 있다.

+ 사용되지 않는 동적 메모리를 명시적으로 반환 해야한다.

* void free (void \*memblock);

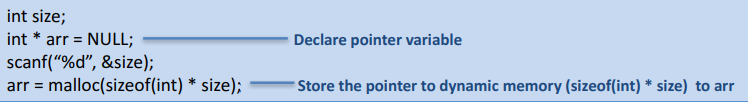
+ 동적 메모리가 할당 해제 된 후 포인터 변수를 NULL로 설정하는 것이 좋다.

(NULL로 할당 되지 않은 포인터 변수를 표시한다.)



* Dynamic Memory Usage

동적으로 할당된 메모리를 가리키는 포인터 변수는 배열로 사용 될 수 있다.

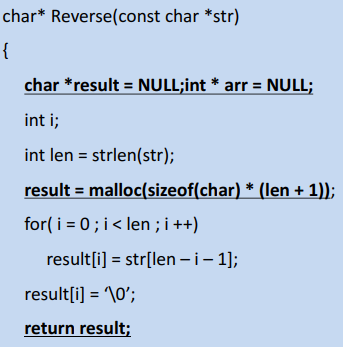




* Scope and Life-time of Dynamic Memory

함수 내에 할당 된 동적 메모리는 함수가 존재 한 후에도 여전히 살아 있다

따라서, 함수가 존재하더라도 프로그램문이 free()에 의해 할당 해제가 되지 않는 한 메모리 공간에 대한 포인터를 사용할 수 있으며, 동적 메모리 공간을 사용 할 수 있다.



Reverse()함수가 종료된 후에, 동적메모리 공간이 할당 해제되지 않았다.

