190409

**\*Name Hiding 예시**

int apple = 5;  
{

Int apple = 1;

}

//다음과 같이 이미 선언된 변수를 더 작은 영역 안에서 한번 더 선언해서 초기화 했을 경우 이전에 선언한 변수는 이름을 숨겨 버린다. 이를 Name Hiding이라고 한다.

* 이와 같은 케이스에서 가급적이면 그냥 다르게 변수이름을 짓는 게 낫다.

Tip) Namespace의 Nest Depth는 최대 3까지만 하자.

* 3중첩은 거의 못 봄
* 2중첩도 가끔 씀
* 1중첩은 자주 쓰는 정도…

로 알아두자.

\*Name Hiding된 변수 사용하기

예시 1)

Int a = 123; //다음에 나올 선언에 의해 Name Hiding 된다.

{

Int a = 1;

}

//다음 과 같은 케이스에서 123을 출력하는 변수 a를 출력하고 싶다면 :: 연산자를 사용한다.

cout<<::a<<endl;

cout<<a<<endl;

**\*Global Values, Static Values, Internal Linkage, External Linkage**

(1)Static Values란?

* Static의 의미는 OS로부터 받은 메모리가 정적(고정됨)이라는 뜻으로 Scope를 벗어나도 프로그램상에서 계속 메모리 반납 하지 않고 지속적으로 사용할 수 있게 된다.
* 초기화를 딱 한번 밖에 안 한다.(**초기화를 반복 하지 않는다**)
* 그러므로, static변수의 선언 시 초기화는 반드시 해줘야 한다.

**\*Global Values**

Q. 전역변수가 위험한 이유

A. 전역변수를 바꾸는 곳이 다른 cpp파일일 경우 의도한대로 코딩이 안되고 디버깅도 힘들다.

Q. 전역 변수를 사용함에 따르는 애로사항을 해결하는 방법들

* 함수의 argument로 사용 될 때 지역변수로 초기화하고 Parameter를 이용해 전달해주는 방식을 사용.
* 전역변수의 이름을 G\_(이름) 으로 하여 전역변수임을 알기 쉽게 명시해두는 방식.
* 어찌 됐던 간에 전역변수는 최대한 안 쓰는 것이 좋다.

Q. Global Values와 Static Values의 차이

A. Global Values는 다른 cpp파일에서 쓰일 수 없다.

\*Internal Linkage

cpp파일 안에서만

int g\_x; //external linkage

static int g\_x; //internal linkage

const int g\_x; //안됨

extern int g\_z;

extern const int g\_z; //가능함. 대신에 어느 한 곳에서만 초기화를 해줬어야 함.

int g\_y(1);

static int g\_y(1); //다른 CPP파일에서 접근이 불가능한 변수 Static Global Linkage

const int g\_y(1);

//같은 파일 안에서만 접근하는 변수

extern int g\_w(1); // 초기화(해줬기 때문에 다른곳에서 초기화X), 전역변수, 다른cpp파일에서 접근 가능

extern const int g\_w(1); //상수, 초기화 필수, 외부(CPP파일 끼리)에서 접근 가능함.