1. 이름공간(namespace)

* 특정 이름 공간의 특정 함수만 사용할 경우

Ex) using NP::func; (NP 이름공간 안에 있는 func()를 기본적으로 사용할 것이다)

* 특정 이름 공간의 함수를 모두 사용할 경우

Ex) using namespace NP (NP 이름공간 안에 있는 모든 함수를 사용할 것이다.)

* 이름이 없는 이름공간을 사용할 경우 이름공간 내에서의 변수,함수는 이름공간 내에서만 사용 가능하다.

씹어먹는 C++ 강좌의 저자는 using namespace 사용을 지양하고 std::cout 처럼 이름공간을 명시하여 같은 이름 충돌을 보호하는 것이 좋다고 한다. std와 같은 이름공간에는 수많은 함수가 있기 때문에.

1. 변수명 짓기 TIP

* 낙타식(NumOfPeople)이나 언더바(num\_of\_people) 둘 다 가능하지만 혼합하면 안됨. 변수명이 길어져도 좋으니 축약하는 것보단 이해하기 쉽게 만드는 것이 중요.

1. 레퍼런스(참조, &)

* 정의할 때 반드시 초기화 해야함, 주소를 저장하는 개념이 아니라 ‘별명’처럼 초기화된 변수와 동일한 취급을 함. 변수처럼 메모리를 따로 차지하지 않음.
* 레퍼런스는 기본적으로 상수를 참조할 수 없으나 const 키워드를 붙이면 가능하다.
* 레퍼런스의 레퍼런스, 레퍼런스의 배열, 레퍼런스의 포인터는 존재할 수 없다.

Ex) const int &num = 5; // const를 빼면 불가능

* 배열의 레퍼런스는 가능하다.

Ex) int (&ref)[3][2] = arr; // [3][2]크기를 가지는 배열의 레퍼런스

1. 변수 선언 TIP

* 바뀌지 않을 값에 const를 넣어 상수로 선언하는 습관을 들이자.

1. 포인터

* 배열의 이름은 배열의 0번째 원소를 가리키는 상수 포인터다. 상수 포인터란 const 키워드 위치에 따라 값이나 주소를 수정할 수 없는 포인터를 뜻한다.

1. 메모리 할당

* 배열의 크기(메모리)는 컴파일 시간에 확정되야한다.
* Heap은 사용자가 자유롭게 할당하거나 해제할 수 있는 메모리의 부분이다.
* C언어에서의 동적 할당은 malloc/free, C++언어에서의 동적 할당은 new/delete