함수 객체 (Function Object, Functor)

- 최초에는 "() 연산자를 재정의 해서 함수 처럼 사용 가능한 객체" 라는 의미.

- 요즘에는 "() 연산자를 사용해서 함수 처럼 호출 가능한 모든 객체" – 함수 포인터, ()를 재정의한 클래스, 멤버 함수 포인터등…

함수 객체의 장점

- 알고리즘에 전달 시 일반 함수는 인라인 치환이 안되지만 함수 객체는 인라인 치환 가능. – "C++ Intermediate(중급)" 과정 참고

- 상태를 가질 수 있다.

STL 이 제공하는 함수 객체

- <functional> 헤더

산술연산	plus<>
	minus<>
	multiplies<>
	divides<>
	modulus<>
	negate<>
비교연산	equal_to<>
	not_equal_to<>
	greater<>
	less<>
	greater_equal<>
	less_equal<>
논리연산	logical_and<>
	logical_or<>
	logical_not<>

STL 알고리즘은 함수를 인자로 가지는 경우 가 많이 있다.

- 알고리즘의 활용도를 더욱 높여 준다.
- for_each, transform.

단항 함수	: 인자가 한 개인 함수
이항 함수	: 인자가 두 개인 함수

- 일반 함수 뿐 아니라, 함수 객체, 람다 표현식을 사용 할 수 있다. - ()로 호출 가능한 모든 객체를 사용 가능.

람다 표현식(Lambda Expression)

- C++11 에서 추가된 문법
- 익명의 함수 객체를 만드는 표현식
- “[]“ : 람다 표현식이 시작 됨을 알리는 “lambda introducer”

transform

- 인자로 전달된 컨테이너의 모든 요소에 연산을 적용 후 다른 컨테이너에 복사하는 알고리즘

- 단항 함수 버전과 이항 함수 버전 제공

- 자세한 함수 모양은 cppreference.com 참고