

# Day-6

## Conversion Rules

### Implicit type conversion rules : 암묵적인 형 변환 규칙

암묵적인 형 변환 규칙은 모든 작업 전에 순서대로 적용됨

⊗: anyoperation(\*,+,/,-,%,etc.)

- Floating point 자동 타입 변환
  - floating\_type ⊗ interger\_type → floating\_type
- Implicit integer 자동 타입 변환
  - small\_integral\_type ⊗ small\_integral\_type → int
- Size 자동 타입 변환
  - small\_type ⊗ large\_type → large\_type
  - 작은 크기의 타입 → 큰 크기의 타입
- Sign 자동 타입 변환
  - signed\_type ⊗ unsigned\_type → unsigned\_type

### Implicit Promotion

32bit 보다 작은 데이터 타입은 signed or unsigned 암묵적으로 int로 변환됨

## auto Declaration

### auto Keyword

- C++11의 auto 키워드는 컴파일러가 변수의 타입을 자동으로 추론하도록 지정함

```

auto a = 1 + 2;    // 1은 int형, 2는 int형 따라서 1 + 2 또한 int
                  // -> a는 int형이 됨
auto b = 1 + 2.0; // 1은 int형, 2.0은 double. 따라서 1 + 2.0은 double
                  // -> b는 double형이 됨

```

- auto는 기능 유지 및 복잡한 타입 정의를 숨기는데 매우 유용함
- 반면 auto를 통해 타입을 과도하게 숨기면 가독성이 떨어질 수 있음

## auto Keyword - Function

- C++11/ C++14에서는 auto를 사용하여 함수 출력형을 정의 할 수 있음

```

auto g(int x) -> int { return x * 2;} // C++11
auto h(int x) { return x * 2;} // C++14

```

- C++20에서는 auto를 사용하여 함수 입력을 정의 할 수 있음

```

void f(auto x){}
f(3);    // 'x'는 int
f(3.0);  // 'x'는 double

```