C++: Rules for Different Ways of Initialization

		always has defined value	narrowing is error	works for `initializer_list<>`	`explicit` conversion supported	works for aggregates	works for `auto`	works for members
	Type `i;`	no	-	no	-	✓ (no init)	no	✓
	Type `i{};`	✓	-	✓	-	✓	no	✓
	Type `i();`	**function declaration**						
direct initialization	*Type* `i{x};`	✓	✓[1]	✓	✓	✓	✓[2]	✓
	Type `i(x);`	✓	no	no	✓	since C++20, not nested	✓	no
	Type `i(x, y);`	✓ (2 args)	no	no	✓	since C++20, not nested	✓	no
copy initialization	*Type* `i = x;`	✓	no	no	no	no	✓	✓
	Type `i = {x};`	✓	✓[1]	✓	no	✓	✓ init-list	✓
	Type `i = (x);`	✓ (1 arg)	no	no	no	since C++20, not nested	✓ (1 arg)	✓ (1 arg)
	Type `i = (x, y);`	✓ (last arg)	no	no	no	since C++20, not nested	✓ (last arg)	✓ (last arg)

[1]: **`g++`** needs **`-pedantic-errors`** or **`-Werror=narrowing`** to detect narrowing errors

[2]: **`std::initializer_list<>`** before g++ 5, clang 3.8, and Visual Studio 2015

josuttis.com/cpp/c++initialization.pdf

@NicoJosuttis