Type Deduction (Cheat Sheet)

C++98 only required type deduction for template parameters

C++11 introduced – auto, decltype, universal references and lambda (return or capture)

C++14 introduced – decltype(auto) and lambda (auto capture)

Template Type Deduction

Template<typename T> void fun(T param) // By value

Input	T – type	Param - type
int	int	int
int&	int	int
const int	int	int
const int&	int	int

Auto

auto val = input; // By value

Input	auto – type	description
int	int	
int&	int	Reference removed
const int	int	const removed
const int&	int	const/Reference removed

auto& val = input; // By value

Input	auto – type	val - type
int	int	int &
int &	int	int &
const int	const int	const int &
const int&	const int	const int &

std::initializer_list<> and auto

auto x { 4, 3, 2, 1}; // x of type std::initializer_list<int>

auto x = { 4, 3, 2, 1}; // x of type std::initializer_list<int>

Template Type Deduction

template<typename T> void fun(T& param) // Reference

Input	**T – type**	**Param - type**
int	int	int&
int&	int	int &
const int	const int	const int &
const int&	const int	const int &

template<typename T> void fun(const T& param) // Reference to const

Input	**T – type**	**Param - type**
int	int	const int&
int&	int	const int &
const int	int	const int &
const int&	int	const int &

template<typename T> void fun(T* param) // Pointer

Input	**T – type**	**Param - type**
int*	int	int*
const int*	const int	const int *

template<typename T> void fun(T&& param) // Universal reference

Input	**T – type**	**Param - type**
int	int&	int&
Int&	int&	int&
const int	const int&	const int &
const int&	const int&	const int &
temp(int)	int	int&& *(rvalue)*

lValue -> lvalue reference

decltype

decltype(name) val

Named variable	decltype(name)	description
int	int	
const int &	const int&	

Return Type Deduction

Example	Return type
auto fun() { int ans = 42; return ans;}	int
decltype(auto) fun() { int ans = 42; return ans;}	Int
decltype(auto) fun() { int ans = 42; return (ans);}	int&

Move and Forward

Move and Forward do not have any 'executable' code – simple casts

Move – unconditional cast to 'rvalue'

Forward – conditional cast to 'rvalue'

