Neue C++ Features ab C++03

Inhaltsverzeichnis

[C++03: 2](#_Toc55376790)

[- New features 2](#_Toc55376791)

[C++11: 2](#_Toc55376792)

[- Features that were merged into C++11 2](#_Toc55376793)

[- Core language features 2](#_Toc55376794)

[- Library Headers: 3](#_Toc55376795)

[- Library features 3](#_Toc55376796)

[C++14 4](#_Toc55376797)

[- New language features 4](#_Toc55376798)

[- New library features 4](#_Toc55376799)

[C++17 5](#_Toc55376800)

[- features that were merged into C++17 5](#_Toc55376801)

[- Obsolet/removed features 5](#_Toc55376802)

[- New language features 6](#_Toc55376803)

[- New library headers 6](#_Toc55376804)

[- New library features 7](#_Toc55376805)

[C++20 8](#_Toc55376806)

[- new language features 8](#_Toc55376807)

[- new headers: 9](#_Toc55376808)

[- Library features: 9](#_Toc55376809)

[Quellen 10](#_Toc55376810)

# C++03:

* Eher eine kleine Revision, die technische Fehler aus der Version 98 beheben sollte.
* New features:
  + Value initialization
    - This is the initialization performed when a variable is constructed with an empty initializer.

# C++11:

* Die zweite wichtige Version nach C++98 (die nächste ist C++17). Große Anzahl an Veränderungen.
* Features that were merged into C++11**:**
  + All of TR1 except Special Functions
  + From Boost: The thread library, exception\_ptr, error\_code and error\_condition, **iterator improvements** (std::begin, std::end, std::next, std::prev)
  + C-style Unicode conversion functions
* Core language features**:**
  + auto and decltype
  + defaulted and deleted functions
  + final and override
  + trailing return types
  + rvalue references
  + move constructors and move assignement operators
  + scoped enums
  + constexpr and literal types
  + list initialization
  + delegating and inherited constructors
  + brace-or-equal initializers
  + nullptr
  + long long
  + char16\_t and char32\_t
  + type aliases
  + variadic templates
  + generalized (non-trivial) unions
  + generalized PODs (trivial types and standard-layout types)
  + Unicode string literals
  + user-defined literals
  + attributes
  + lambda expressions
  + noexcept specifier and noexcept operator
  + alignof and alignas
  + multithreaded memory model
  + thread-local storage
  + GC interface
  + range-for (based on a Boost library)
  + static\_assert (based on a Boost library)

## Library Headers:

* + <typeindex>
  + <type\_traits>
  + <chrono>
  + <initializer\_list>
  + <tuple>
  + <scoped\_allocator>
  + <cstdint>
  + <cinttypes>
  + <system\_error>
  + <cuchar>
  + <array>
  + <forward\_list>
  + <unordered\_set>
  + <unordered\_map>
  + <random>
  + <ratio>
  + <cfenv>
  + <regex>
  + <atomic>
  + <thread>
  + <mutex>
  + <future>
  + <condition\_variable>
* Library features**:**
  + atomic operations library
  + emplace() and other use of rvalue references throughout all parts of the existing library
    - std::unique\_ptr
    - std::move\_iterator
  + std::initializer\_list
  + stateful and scoped allocators
  + std::forward\_list
  + chrono library
  + ratio library
  + new algorithms
  + Unicode conversion facets
  + thread library
  + std::exception\_ptr
  + std::error\_code and std::error\_condition
  + iterator improvements:
    - std::begin
    - std::end
    - std::next
    - std::prev
  + Unicode conversion functions

# C++14

* Minor revision of the C++ standard
* New language features**:**
  + variable templates
  + generic lambdas
  + lambda init-capture
  + new/delete elision
  + relaxed restrictions on constexpr functions
  + binary literals
  + digit separators
  + return type deduction for functions
  + aggregate classes with default non-static member initializers.
* New library features**:**
  + std::make\_unique
  + std::shared\_timed\_mutex and std::shared\_lock
  + std::integer\_sequence
  + std::exchange
  + std::quoted
  + and many small improvements to existing library facilities, such as
    - two-range overloads for some algorithms
    - type alias versions of type traits
    - user-defined literals for basic\_string, duration and complex
    - etc.

# C++17

* major revision of the C++ stadnard after C++11
* features that were merged into C++17:
  + From TS's:
    - the filesystem library
    - the library fundamentals v1:
      * optional
      * any
      * string\_view
      * polymorphic allocators
      * searchers
      * apply
    - parallelism v1:
      * execution policies
      * reduce
      * inclusive\_scan
      * exclusive\_scan
    - From special function IS:
      * mathematical special functions
    - from library fundamentals v2:
      * std::gcd
      * std::lcm
* Obsolet/removed features:
  + auto\_ptr
  + deprecated function objects
  + std::random\_shuffle
  + std::unexpected
  + the obsolete iostreams aliases
  + trigraphs
  + the register keyword
  + bool increment
  + std::iterator
  + std::raw\_storage\_iterator
  + std::get\_temporary\_buffer
  + std::is\_literal\_type
  + std::result\_of
  + all of <codecvt>
* New language features:
  + fold-expressions
  + class template argument deduction
  + non-type template parameters declared with auto
  + compile-time if constexpr
  + inline variables
  + structured bindings
  + initializers for if and switch
  + u8 character literal
  + simplified nested namespaces
  + using-declaration declaring multiple names
  + made noexcept part of type system
  + new order of evaluation rules
  + guaranteed copy elision
  + lambda capture of \*this
  + constexpr lambda
  + attribute namespaces don't have to repeat
  + new attributes:
    - [[fallthrough]]
    - [[nodiscard]]
    - [[maybe\_unused]]
  + \_\_has\_include
* New library headers:
  + <any>
  + <optional>
  + <variant>
  + <memory\_resource>
  + <string\_view>
  + <charconv>
  + <execution>
  + <filesystem>
* New library features**:**
  + Utility types
    - tuple:
      * apply
      * deduction\_guides
      * make\_from\_tuple
    - variant
    - as\_const
    - searchers
    - optional
    - any
    - not\_fn
  + memory management
    - uninitialized memory algorithms
      * destroy\_at
      * destroy
      * destroy\_n
      * uninitialized\_move
      * uninitialized\_value\_construct
    - weak\_from\_this
    - memory\_resource and polymorphic\_allocator
    - aligned\_alloc
    - transparent owner\_less
    - array support for shared\_ptr
    - allocation functions with explicit alignment
  + compile-time programming
    - byte
    - conjunction/disjunction/negation
    - type trait variable templates (xxx\_v)
    - is\_swappable
    - is\_invocable
    - is\_aggregate
    - has\_unique\_object\_representations
  + algorithms
    - clamp
    - parallel algorithms and execution policies
    - reduce
    - inclusive\_scan
    - exclusive\_scan
    - [gcd](https://en.cppreference.com/w/cpp/numeric/gcd)
    - lcm
  + iterators and containers
    - map/set extract and map/set merge
    - map/unordered\_map try\_emplace and insert\_or\_assign
    - contiguous iterators
    - non-member size/empty/data
  + numerics
    - mathematical special functions
    - 3D hypot
  + Other
    - launder
    - to\_chars/from\_chars
    - is\_always\_lock\_free
    - scoped\_lock
    - cache line interface
    - uncaught\_exceptions
    - timespec\_get
    - rounding functions for duration and time\_point

# C++20

* current version of C++ standard
* new language features**:**
  + Feature test macros
  + 3-way comparison operator <=> and operator==() = default
  + designated initializers
  + init-statements and initializers in range-for
  + char8\_t
  + [[no\_unique\_address]]
  + [[likely]]
  + [[unlikely]]
  + pack-expansions in lambda init-captures
  + removed the requirement to use typename to disambiguate types in many contexts
  + consteval, constinit
  + further relaxed constexpr
  + signed integers are 2's complement
  + aggregate initialization using parentheses
  + Coroutines
  + Modules
  + Constraints and concepts
  + Abbreviated function templates
  + DR: array new can deduce array size

## new headers:

* + <concepts>
  + <coroutine>
  + <compare>
  + <version>
  + <source\_location>
  + <format>
  + <span>
  + <ranges>
  + <bit>
  + <numbers>
  + <syncstream>
  + in Thread support library:
    - <stop\_token>
    - <semaphore>
    - <latch>
    - <barrier>

## Library features:

* + Library feature-test macros
  + Formatting library
  + Calendar and Time Zone library
  + std::source\_location
  + std::span
  + std::endian
  + array support for std::make\_shared
  + std::remove\_cvref
  + std::to\_address
  + floating point atomics, shared\_ptr atomics
  + thread-coordination classes: std::barrier, std::latch, and std::counting\_semaphore
  + std::jthread and thread cancellation classes: std::stop\_token, std::stop\_source, and std::stop\_callback
  + std::osyncstream
  + std::u8string and other char8\_t uses
  + constexpr for <algorithm>, <utility>, <complex>
  + std::string::starts\_with / ends\_with and std::string\_view::starts\_with / ends\_with
  + std::assume\_aligned
  + std::bind\_front
  + std::c8rtomb/std::mbrtoc8
  + std::make\_obj\_using\_allocator etc
  + std::make\_shared\_for\_overwrite/std::make\_unique\_for\_overwrite
  + heterogeneous lookup in unordered associative containers
  + std::pmr::polymorphic\_allocator with additional member functions and std::byte as its default template argument
  + std::execution::unseq
  + std::midpoint and std::lerp
  + std::ssize
  + std::is\_bounded\_array, std::is\_unbounded\_array
  + Ranges
  + uniform container erasure: std::erase/std::erase\_if, e.g. std::erase(std::list) or std::erase\_if(std::map) etc
  + Mathematical constants in <numbers>

# Quellen

<https://en.cppreference.com/w/cpp/language/history>, letzter Zugriff: 4.11.20 10:00 Uhr