

## Index

- digit separator (`'`), 152–156
  - templated variable declarations, 157–166
- `std::index_sequence`, 293
- `<type_traits>` header, 1014
- unsafe features
  - auto** return-type deduction, 1182–1204
  - decltype(auto)** placeholder, 1205–1214
- user-defined literals (UDLs) in, 852–853
- xvalues*, 721–723
- C++17
  - capturing **\*this** by copy, 611n7
  - conditionally supported, 425n7
  - dynamic exception specifications, 1085n1
  - exception specifications and type system, 1089n5
  - false sharing, avoiding, 175n6
  - fold expressions, 955n25
  - guaranteed copy elision, 216n1, 648n11, 805n30, 827n54
  - if **constexpr** language feature, 641n10
  - nested namespaces, 1055n1
  - new keywords in, 1023n7
  - pmr allocators, 763n25
  - polymorphic memory resources, 190n3
  - range-based **for** loops, 681n2
  - sentinels, 707n12
  - `std::any`, 187n2
  - `std::pmr::monotonic_resource`, 468n27
  - `std::pmr::unsynchronized_pool_resource`, 468n27
  - `std::string_view`, 874n1
  - `std::variant`, 452n19, 1180n2
  - structured binding, 201n2, 685n3
  - trivial types, 425n7
  - type traits, 651n12
- C++20
  - bit field initialization, 329n4
  - char**-like object, 479n29
  - concepts, 122n5, 208n3, 480n30, 1201n5
  - constexpr** functions as destructors, 463n25
  - constexpr** functions in algorithms, 294n19
  - constexpr** keyword, 75n5, 304n1, 316n8
  - contracts, 467n26
  - deleted constructors, 247n8
  - designated initializers, 139n1
  - destructors, 407n3
  - encapsulation of helper types, 85n3
  - enumeration comparisons, 335n1
  - floating-point non-type template parameters, 903n7
  - generic lambdas, explicit parameter types, 193–194
  - implicit conversion, 223n3
  - implicitly movable entities, 735n13
  - manifestly constant evaluated, 258n1
  - moves in **return** statements, 740n16
  - nested namespaces, 1055n1
  - new keywords in, 1023n7
  - `[[no_unique_address]]` attribute, 1029n15
  - Ranges Library, 686n4, 687n5
  - ranges library, 391–393
  - relaxed restrictions on **constexpr** functions, 960n1
  - reordering data members, 178n10
  - requires clause, 486n31
  - sentinels, 707n12
  - Standard Library-related restrictions, 1078n6
  - `std::bit_cast`, 514n41, 516n42
  - `std::is_constant_evaluated()`, 297n20
  - `std::is_pod`, 438n14
  - `std::remove_cvref<T>`, 399n6
  - terse concept notation syntax, 398n5
  - trivially destructible types, 430n9
  - typename** disambiguator, 382n1
  - unscoped enumerated types, 833n2
  - user-declared constructors, 274n7
- C++23
  - guaranteed copy elision, 805n30
  - reordering data members, 178n10
- C++-only types, translating to C, 452–456
- C99, flexible array members, 404n1
- cache associativity, 182n11
- cache hit, 181
- cache lines, 174–175, 181–183, 459, 1142
- cache miss, 182
- call operators in functor classes, 574–575
- callable objects, 70, 994
- callback functions, 669. *See also* lambda expressions
- callbacks, event-driven, 603–604
- capture default, 582–583, 600, 608
- captured by copy, 582, 611–612, 990–992
- captured by reference, 582
- captured by value. *See* captured by copy
- captured variables, 582–585, 590–591, 602, 609–610, 990
- `[[carries_dependency]]` attribute
  - description of, 998–1000
  - further reading for, 1006
  - potential pitfalls, 1005
  - use cases, 1000–1005
- Carruth, Chandler, 1134
- carry dependency, 999
- cast, 345
- character literals, 837, 844n1
- char**-like object, 479, 479n29
- `checkBalance` function, 15
- `checksumLength` function, 27, 28n1