Index

  - value initialization, avoiding the most vexing parse, 237–238
- arithmetic operators, braced lists and, 254–255
- arithmetic types
  - **`enum`** class, 334, 337–339
  - implicit conversion, avoiding, 337–339
- array types
  - **`alignof`** operator, 184
  - as literal types, 280
  - as standard-layout types, 417
  - as trivial types, 425
- arrays
  - built-in, deducing, 211–212
  - initialization with `std::initializer_list`
    - annoyances, 567–571
    - description of, 553–561
    - further reading for, 571
    - potential pitfalls, 566–567
    - range-based **`for`** loops, 571–572
    - use cases, 561–566
  - size deduction, lack of, 330
  - traversing with range-based **`for`** loops, 683–684
- array-to-pointer decay, 220, 222
- ASCII
  - basic source character set, 130
  - Unicode string literals, 129
- as if, 307
- assert, 656
- assert statements in dependency chain, 1002
- assignable (type), 486
- assignment operator, 521–522
  - braced lists and, 254–255
  - lvalue references, 816
- atomic (operation), 80–82
- attribute lists, 922
- attribute support. *See also* attributes
  - description of
    - attribute placement, 13
    - attribute syntax, 12–13
    - standardized compiler-specific attributes, 13–14
  - potential pitfalls
    - undefined behavior, 19
    - unrecognized attributes, 18–19
  - use cases
    - control of external static analysis, 17–18
    - hints for additional optimization opportunities, 15–16
    - prompting of compiler diagnostics, 14–15
    - statement of explicit assumptions, 16–17
    - statements of semantic properties, 18
- attributes. *See also* attribute support
  - `[[carries_dependency]]`
    - description of, 998–1000
    - further reading for, 1006
    - potential pitfalls, 1005
    - use cases, 1000–1005
  - `[[clang::no_sanitize]]`, 14
  - definition of, 12
  - `[[deprecated]]`, 14
    - description of, 147–148
    - potential pitfalls, 150
    - use cases, 148–150
  - `[[gnu::cold]]`, 15
  - `[[gnu::`**`const`**`]]`, 16–17, 19
  - `[[gnu::pure]]`, 14, 16
  - `[[gnu::warn_unused_result]]`, 14–15, 15n7
  - `[[gsl::suppress]]`, 17–18
  - `[[noreturn]]`, 13
    - description of, 95
    - further reading for, 98
    - potential pitfalls, 97–98
    - use cases, 95–97
- **`auto`** variables
  - annoyances, 212–213
    - nonstatic data members, not allowed, 212
    - template argument deductions, not all allowed, 212–213
  - braced initialization and, 253–254
  - **`decltype`**`(`**`auto`**`)` placeholder
    - annoyances, 1213
    - description of, 1205–1210
    - potential pitfalls, 1212–1213
    - use cases, 1210–1212
  - description of, 195–199
  - further reading for, 214
  - idioms for, 1213
  - potential pitfalls, 204–212
    - compromised readability, 204
    - deducing built-in arrays, 211–212
    - deduction for list initialization, 210–211
    - hidden properties of fundamental types, 209–210
    - interface restrictions, lack of, 208–209
    - unexpected conversions, 206–208
    - unintentional copies, 204–206
  - return-type deduction
    - annoyances, 1201–1203
    - description of, 1182–1194
    - potential pitfalls, 1200
    - use cases, 1194–1200
  - use cases, 200–203
    - deeply nested variable types, 202–203
    - ensuring variable initialization, 200
    - implementation-defined or compiler-synthesized variable types, 202
    - preventing unexpected implicit conversions, 201