

## Index

- derived classes
    - compile-time visitation, 1050–1054
    - preventing with **final** contextual keyword, 1007, 1014–1015
  - design patterns, 669
  - designated initializers, 139n1
  - destructive move, lack of, 811–812
  - destructors
    - in C++20, 407n3
    - as **constexpr** functions, 463n25
    - declaring special member functions, 34
    - exception specifications and, 1086
    - final** contextual keyword, 1008
    - noexcept** by default, 653–654
    - rvalue references, 752
    - skippable, 464–470
    - user-provided, 755–757
    - vertical encoding, 450
  - devirtualize, 1011
  - diagnostics, compiler, 14–15
  - diffusion, 183n14
  - digit separator ('). *See also* binary literals
    - description of, 152–153
    - further reading for, 154
    - loss of precision in floating-point literals, 154–156
    - use cases, 153
  - dimensional unit types, 863–865
  - direct aggregate initialization, 493
  - direct braced initialized, 455
  - direct initialization, 215, 754
    - explicit conversion operators, 62
    - in factory functions, 240
    - in generic code, 239
    - of members, 241–242
    - for nonstatic data members, 318
    - syntax, 328–329
  - direct initialized, 230
  - direct list initialization, 228–231
    - copy list initialization, compared, 231–232
    - in factory functions, 240
    - in generic code, 239
    - of members, 241–242
    - for nonstatic data members, 318
    - `std::initializer_list`, 555
  - direct mapped, 182n11
  - disabling
    - implicit moves, 244–246
    - named return-value optimization (NRVO), 783–784
    - NRVO, 244–246
  - disambiguators, 28–30
  - discriminated unions, 937–948, 1177–1180
  - divide and conquer, 297
  - documentation of default values, 325
  - double-checked-lock pattern (C++03), 81–82
  - duck typing, 1052
  - dumb data, 668n7
  - duplicate names, loss of access in namespaces, 1056–1058
  - dynamic binding, 1015
  - dynamic dispatch, 1011
  - dynamic exception specifications, 618–619, 1085, 1089, 1090
    - compatibility with **noexcept** specifications, 621
    - noexcept** exception specification, compared, 1101–1102
    - violating, 1093
  - dynamic types, 416
- ### E
- EBCDIC, 129n1
  - Effective C++* (Meyers), 3
  - elaborated type specifiers, 1031–1032
  - embedded development, 145
  - embedded systems, 1101
  - embedding code in C++ programs, 111–112
  - emplacement, 390–391
  - empty-base optimization (EBO), 185, 499, 607, 933, 1028–1030
  - encapsulation
    - of helper types, 85n3
    - of implementation details, 343–344
    - opaque enumerations, 663
    - types within functions, 84–85
  - encoding prefixes, 844
  - entities
    - decltype** use with, 25–26
    - [[deprecated]] attribute, 147–150
  - enum** class
    - annoyances, 351
    - description of, 332–337
    - further reading for, 352
    - potential pitfalls, 344–350
      - bit flags, 347–348
      - collections of named constants, 346–347
      - external use of opaque **enumerators**, 350
      - iteration, 348–350
      - strong typing can be counterproductive, 344–346
    - scoped enumerations, 335–336
    - underlying types (UTs) and, 337
    - unscoped C++03 enumerations, work-arounds for, 332–333
    - use cases, 337–344
      - encapsulating implementation details, 343–344
      - implicit conversion to arithmetic types, avoiding, 337–339