

Index

literals (cont.)

- digit separators (‘) in
 - description of, 152–153
 - further reading for, 154
 - loss of precision in floating-point literals, 154–156
 - use cases, 153
- floating-point, 154–156, 837, 869–870
- integer, 837, 869–870
- raw string
 - description of, 108–111
 - potential pitfalls, 112–114
 - use cases, 111–112
- Unicode
 - description of, 129–130
 - potential pitfalls, 130–132
 - use cases, 130
- user-defined
 - annoyances, 869–871
 - description of, 835–853
 - further reading for, 872
 - potential pitfalls, 867–869
 - use cases, 853–867
- little-endian **float** layouts, 531–534
- local declarations, 662, 675–677
- local functions, emulating, 598–599. *See also* lambda expressions
- local scope. *See* block scope
- local variables in unevaluated contexts, 610–611
- locality of reference, 181, 742, 773n26
- local/unnamed types. *See also* **decltype**; lambda expressions
 - description of, 83–84
 - use cases, 84–87
 - configuring algorithms via lambda expressions, 86–87
 - encapsulating types within functions, 84–85
 - instantiating templates with local function objects as type arguments, 85–86
- loggedSum function, 28, 31
- logical optimization, 365
- logical or (||) operator, 265
- long long** integral type
 - description of, 89
 - further reading for, 92
 - potential pitfalls, 91–92
 - use cases, 89–91
- long** type, relative size of, 91–92
- long-distance friendship, 1035–1036, 1041
- loops. *See* range-based **for** loops
- lossy conversions, restrictions on, 222–224
- low-level value-semantic types (VSTs), creating, 742–751

- lvalue references, 26, 716, 1118, 1133
 - in C++11/14, 717–720
 - declarations prior to C++11, 815–818
 - evolution of, 807, 813–828
 - forbidding operations on, 1165–1167
 - implicit moves in **return** statements, 735–737
 - range-based **for** loops, 703
 - rvalue references, introduction to, 710–711
- lvalue-to-rvalue conversion, 501

M

- macro-defined namespaces, 1083–1084
- macro-invocation syntax, 248–249
- macros. *See also* functions
 - alternatives to, 286–287
 - offsetof
 - aggressive usage, 520–521
 - navigating compound objects, 456–460
 - POD type usage, 410–412
 - support for, 423–425
- magic constants, 308
- managed allocators, 1021–1022
- mandatory RVO, 807n31
- mangled names, 1056, 1114n24
- manifestly constant evaluated, 258n1
- mantissa, 155
- materialization, 717
- materialize, 1163n1
- maximal fundamental alignment, 193
- mebibyte conversion, 286–287
- mechanisms, 51
- member functions
 - constexpr** as implicitly **const**-qualified, 300
 - hiding, 56–57
 - overriding, 105–106
 - variadic member functions, 892–894
- member initialization lists, 230
- member initializer lists
 - copy list initialization in, 249–250
 - delegating constructors, 46
 - nonstatic data member initialization, 318
 - pack expansion, 917–918
- member initializers, default
 - annoyances, 328–330
 - applicability limitations, 329
 - array size deduction, lack of, 330
 - loss of aggregate status, 330
 - loss of triviality, 329–330
 - parenthesized direct-initialization syntax, lack of, 328–329
 - description of, 318–321