Index

potential pitfalls, 326–328
inconsistent subobject intialization, 326–328
loss of insulation, 326
safety of, 6
union interactions, 320–321
use cases, 322–325
boilerplate repetition, avoiding, 323–325
documentation of default values, 325
nonstatic data member initialization, 322–323
simple **struct** initialization, 322
`memcpy`. *See* `std::memcpy`
memory allocation, 75n4, 181–183
in C++11, 763n25
monotonic, 190–193, 1021–1022
secure buffers, 460–462
memory barriers, 80n7
memory diffusion, 628, 788
memory leak, 74
memory models, synchronization paradigms for, 998
`memory_order_acquire`, 1005n2
`memory_order_consume`, 1005n2
`memory-fence instructions`, 999–1000
metafunctions, 469, 963
forwarding references, 381
requirements in constraints, 398–400
`std::remove_cvref<T>`, 399n6
metaparameters, 948
metaprogramming, 876, 963–964
metaprograms, 257
Meyers, Scott, 3
Meyers singleton, 71–75
microbenchmarks, 1137–1141
mixed-mode builds, 1073
mix-ins, reusable functionality through, 545
mocking, 1017–1020
mocks, 1017–1020
modifiable *rvalue*s, 820–821
modules, 85n3, 1041
monotonic allocators, 1021–1022
monotonic memory allocation, 190–193
Moore's law, 93n5
most vexing parse, avoiding, 237–238
move assignable, 524
move assignment, 750, 756
move construction, 750
move constructors
literal types and, 281
noexcept operator and, 653–654
rvalue references, 710, 714, 732–733
RVO and NRVO requirements, 804–805
`std::list`, 1114
as trivial, 437
user-provided, 760
move operations
avoiding, 183n14
deleted functions, 53
destructive move, lack of, 811–812
enabling with `std::forward<T>`, 395
noexcept operator, 627–631, 658–659
on noncopyable types, 788–791
nonthrowing, 1094–1097
objects into closure, 988–989
as optimization of copying, 741–767
rvalue references, 710, 714–715
some equivalent to copies, 788
throwing in, 787
wrappers for **noexcept**, 1099–1101
move semantics
necessity of, 821–823
rvalue references, 710, 715–716
move-assignment operator
rvalue references, 710, 714, 733
user-provided, 760–761
moved-from objects
inconsistent expectations, 794–803
overly strict requirements, 807–811
rvalue references, 714–715, 788, 807–812
moved-from state, 789, 791–803
move-only types, 570, 641, 644
implementing without `std::unique_ptr`, 791–794
rvalue references, 716, 768–771, 790
moving iterators, return types of, 1211–1212
MSVC
auto redeclaration, 1209
compiler warnings, 150
deduced parameters, 972n1
incompatibly specified alignment, 177
reducing code size, 1104n16, 1111
stack unwinding, 621n4
standardized compiler-specific attributes, 14
trivial copy/move constructors, 528n62
underspecifying alignment, 176
multiple arguments
constraining, 983–984
passing to explicit constructors, 250–252
multiple parameters, handling, 386–388
multiple **return** statements, 1185–1187
multithreaded programs, avoiding false sharing, 174–175
multithreading context, 68, 70–71
mutable closures, 969–970
mutable state, providing for closure, 989–990
`myRandom` function, 19