

Index

constexpr variables (cont.)	contract guarantees
potential pitfalls, 314	nofail functions, 1117–1122
use cases, 307–314	overly strong, 1112–1116
alternative to enumerated compile-time	contract violations, 485
integral constants, 307–310	contracts, 467n26, 485
diagnosing undefined behavior at compile	constexpr functions as part of, 261–262
time, 312–314	new operator, 616
nonintegral symbolic numeric constants,	overly restrictive, 480–482
310–311	rvalue references, 714
storing constexpr data structures, 311-	control constructs
312	emulating, 599–600
constinit keyword, 75n5, 304n1, 316n8	in lambda expressions, 600–601
const-qualified member functions, 300	controlling constant expressions, 285
constraining	conventional string literals, 113
deduced parameters, 970–973	conversion operators
multiple arguments, 983–984	explicit
constructors. See also copy constructors; move	description of, 61–63 potential pitfalls, 66–67
constructors	
boilerplate code repetition, avoiding, 323-	use cases, 63–65
325	as placeholders, 1193–1194 converting constructors, 61
code duplication, avoiding, 48–50	
delegating	cooked UDL operators, 841, 843-845, 870 cookies, 669-675
description of, 46–48	copy assignable, 485–486
potential pitfalls, 50– 51	copy assignment, 485, 758
use cases, 48–50	copy assignment operator
deleted in aggregates, 247	deleted functions, 54
explicit, passing multiple arguments, 250–	rvalue references, 714
252	user-provided, 759
inheriting	vertical encoding, 451
annoyances, 549–552	copy constructible, 455
description of, 535–539	copy construction, 489–492
potential pitfalls, 546–549	copy constructors
use cases, 539–545	declaring special member functions, 34
restrictions on, 269–276	deleted functions, 54
for std::initializer_list, inadvertently	hijacking with perfect-forwarding construc-
calling, 242–244	tor, 395–397
as trivial, 437	literal types and, 281
user-declared, 274n7, 1087	rvalue references, 714
value initializing arguments, avoiding the	RVO and NRVO requirements, 804–805
most vexing parse, 237–238	as trivial, 437
containers	user-provided, 758–759
initialization, 561–562	vertical encoding, 450
iterating all elements, 684–685	copy elision, 390
nested, 22	copy initialization, 215–216
contextual keywords, 1007. See also keywords	in aggregate initialization, 221
override	in generic code, 239
description of, 104–105	for nonstatic data members, 318
further reading for, 107	scalar type, 235–236
potential pitfalls, 106	unions, 506
	copy list initialization, 226–228
use cases, 105–106	direct list initialization, compared, 231–232
potential pitfalls, 1023	in factory functions, 240
contextually convertible to bool , 63–65, 1129	in generic code, 239
continuous refactoring, 147	in member initializer lists, 249–250