

Index

Note: Bold page numbers indicate major topics; italic page numbers indicate examples.

-> 145
 [] 491
 >> 206

A

ABC 507
 about the book 1
 abstract class **G507**
 access declaration 133
 actual parameter 91
 Adamczyk, Steve 203
 ADL 122, **123**, **G507**
 Alexandrescu, Andrei 258, 364
 AlexandrescuDesign 500
 aliasing 420
 alignof **224**
 __alignof__ **224**
 allocator 52, 284
 angle brackets 10, **G507**
 hack **205**, **G507**
 parsing 128, 205
 anonymous union 143
 ANSI **G507**
 archetype **85**
 ArgSelect<> 463
 argument 90, **104**, **G508**
 conversions 172
 deduction **167**
 see argument deduction
 derived class 295
 for function templates **105**

 for template template parameters 51, **111**, 211
 match 488
 named **285**
 nontype arguments **109**
 type arguments **108**
 versus parameter 90
 argument deduction 12
 for function templates 13
 argument-dependent lookup 122, **123**, **G508**
 array
 as template parameter 59
 conversion to pointer 58, 168
 qualification 351
 Array<> 328
 assignment
 with type conversion 45
 associated class 123
 associated namespace 123
 AusternSTL 500
 automatic instantiation 141
 auto_ptr 394
 avoiding
 deduction **174**

B

back()
 for vectors 21, 24
 baggage 284
 barrier 420
 Barton, John J. 174
 Barton-Nackman trick **174**, 177
 versus CRTP 299

- base class
 - conversion to 494
 - dependent 45, 136, 137
 - duplicate 287, 449
 - empty 289
 - nondependent 135
 - parameterized by derived class 295
 - BaseMem<> 449
 - BCCL 500
 - bibliography **499**
 - binary_function 284
 - bind() 468
 - binder
 - for function objects 457
 - Binder<> 460, **465**
 - BinderParams<> 461
 - bindfp() 470
 - bitset 44
 - Blitz++ 318, 500
 - books **499**
 - bool
 - conversion to 392, 494
 - Boost 500
 - BoostCompose 500
 - BoostSmartPtr 500
 - BoostTypeTraits 501
 - Borland 155, 342
 - bounded polymorphism 238
 - BoundVal<> 459
 - bridge pattern 239
 - by-reference versus by-value 331
 - by-value versus by-reference 331
- C**
- call **418**
 - parameter **13**
 - call argument
 - default **173**
 - callback **417**
 - CargillExceptionSafety 501
 - Cfront 163
 - char*
 - as template argument **40, 209**
 - char_traits 284
 - class 10, 101, **G508**
 - associated 123
 - as template argument **40**
 - definition 480
 - dependent base 136
 - inner 227
 - local 101
 - name injection 126
 - nondependent base 135
 - policy class **255**
 - qualification 266, 359
 - template
 - see class template
 - versus struct 87
 - class template **21, 87, G508**
 - as member **45, 95**
 - declaration 22, **95**
 - default argument 30, **103**
 - full specialization **190**
 - overloading 221
 - parameters **173**
 - partial specialization **200**
 - class type 87, **G508**
 - qualification 266, 359
 - class type functor 426
 - client code 245
 - C linkage 99, 435
 - code bloat 201
 - code layout
 - control 224
 - principles 290
 - collection class **G508**
 - see container
 - compiling 74, 153
 - models **141**
 - symbol length 79
 - complete type 143
 - complex 12
 - compose() 447
 - Composer<> 446, **455**
 - composition
 - of function objects 445
 - CompoundT<> 350, 354
 - concept 75, 78
 - constant
 - true constant 515
 - constant-expression 91, 107, **303, G508**
 - const member function **G508**
 - constness
 - and smart pointers 390

contact with the authors **5**
 container **G508**
 as template template parameter 112
 element type 264
 context
 deduced 169
 context-sensitive 119
 conversion
 array to pointer 58, 168
 base-to-derived 494
 derived-to-base 494
 for pointers 494
 for smart pointers 390
 of arguments 172
 sequence 494
 standard 489
 to bool 392, 494
 to ellipsis 107, 489
 to pointer-to-member 392
 to void* 494
 type promotion 271
 user-defined 109, 489
 with templates 45
 conversion-function-id 120
 conversion operator **G508**
 Coplien, James 299
 CoplienCRTP 501
 CoreIssue115 501
 covariant return type 216
 CRTP **295, G508**
 versus Barton-Nackman trick 299
 CSM 279
 <cstdlib> **4**
 curiously recurring template pattern **295, G509**
 CzarneckiEiseneckerGenProg 501

D

debugging 74
 decay 58, 168, 421, **G509**
 of template parameters 102
 declaration 89, **95, 476, G509**
 access 133
 forward 142
 of class template 22
 versus definition 89, **476**
 decorated name 418
 deduced context 169

deduced parameter 14
 deduction 12, **167, G509**
 avoiding **174**
 of initializer 225
 default
 argument
 see default argument
 call arguments **173**
 for template template argument 111
 for template template parameter 111
 nontype parameter 38
 default argument
 for call parameters 97
 for class templates 30, **103**
 for function templates **13, 207**
 definition 9, 89, **476, G509**
 of class types 480
 versus declaration 89, **476**
 demangler 59
 dependent base class 136, **G509**
 dependent name 119, 121, **G509**
 in using-declarations **133**
 of templates **132**
 of types **130**
 deprecated 133
 deque 52
 derivation 135, **285**
 derived class
 as base class argument 295
 design 229
 design pattern 239
 DesignPatternsGoV 501
 determining types 278, **347, 361**
 diagnostics 218
 digraph 127, 129, **G509**
 future 206
 direct call **418**
 directive
 for explicit instantiation 159
 discriminated union 224
 domination rule 289
 dot-C file **61, G509**
 double
 as template argument **40, 210**
 as traits value 252
 zero initialization **56**
 duo **395**

recursive 401
Duo<> 397
duplicate base class 287, 449
dynamic polymorphism **231**, 238

E

EBCDIC 247
EBCO **289**, 449, **G510**
EDG 501
ellipsis 107, 358, 489
EllisStroustrupARM 501
e-mail to the authors **5**
empty()
 for vectors 21
empty base class optimization **289**, 449, **G510**
empty class object 431
enumeration
 qualification 356
 versus static constant 303
error handling 74
error message **75**
example code of the book **5**
exception
 protection 366
 safety 24
expansion
 restricted 174
explicit instantiation **65**, **159**
 directive **G510**
explicit instantiation directive 159
explicit specialization 88, 130, 137, **190**, **G510**
explicit template argument 135
export **68**, **149**, 484
 and inline 69
expression
 constant 107
 for functions 226
expression template **321**, **G510**
 limitations 340
 performance 340
extended Koenig lookup 140

F

feedback to the authors **5**
file organization 61
fixed traits **246**

float
 as template argument **40**, 210
 as traits value 252
 zero initialization **56**
formal parameter 91
forward declaration 142
ForwardParamT 440
friend 101, **113**
 function **114**
 function versus function template 177
 name injection 125, **G510**
 template **117**
full specialization 301, **G510**
 of class templates **190**
 of function templates **194**
 of member templates **197**
func_ptr() 443
function
 call
 see function call
 dispatch table 156
 expression 226
 for types **263**
 object
 see function object and functor
 pointer 421
 qualification 352
 signature 184
 spilled inline 156
 surrogate **496**
 template
 see function template
function call **418**
 operator 417
 syntax 417
function object **417**, **G510**
 and overloading 496
 binders 457
 class type 426
 composition 445
 value binders 457
FunctionPtr<> 441
FunctionPtrT<> 439
function template **9**, 87, **G510**
 argument deduction 13
 arguments **105**
 as member **45**, 95

- declaration **95**
- default call argument 97
- default template argument **13, 207**
- friend **114**
- full specialization **194**
- inline **72**
- nontype parameters 39
- overloading 15, **183**
- partial specialization 213
- versus friend function 177
- functor **417, G510**
 - and overloading 496
 - binders 457
 - class type 426
 - composition 445
 - value binders 457
- FunctorParam<> 438
- fundamental type
 - promotions 274
 - qualification 347
- future **205**

G

- generated specialization 88
- generic programming **240**
- Gibbons, Bill 139, 177, 299
- glossary **507**
- greater 429
- greedy instantiation **155**
- guard
 - for header files 479

H

- Hartinger, Roland 217
- header file **61, G510**
 - <cstdlib> **4**
 - guard 479
 - order 73
 - precompiled **72**
 - <std> 74
 - <stddef.h> **4**
- heterogeneous collection 238
- Hewlett-Packard 139
- higher-order genericity 118
- holder **368**
- home page of the book **5**

I

- identifier 120
- IfThenElse<> **272, 273, 308, 438**
- implicit conversion
 - for smart pointers 390
- implicit instantiation 141
- #include
 - order 73
- include file **G511**
 - see header file
- inclusion model **63, 149**
- index of the book **517**
- indirect call **418, G511**
- induction variables
 - in metaprograms 309
- inheritance 135, **285**
 - domination rule 289
 - duplicate base class 287, 449
- initialization
 - of fundamental types **56**
- initializer **G511**
- initializer deduction 225
- initializer list 56, **G511**
- injected
 - class name 126, **G511**
 - friend name 125
- inline **72**
 - and export 69
- inlining 420
- inner class 227
- Inprise 155
- instance 11, **G511**
- instantiated specialization 88
- instantiation **11, 88, 141, G511**
 - automatic 141
 - diagnostics 218
 - explicit **65, 159**
 - explicit directive 159
 - greedy **155**
 - implicit 141
 - iterated **157**
 - lazy **143**
 - levels 313
 - manual 160
 - mechanisms **141**
 - model 146

- on-demand 141
- point **146**
- queried **156**
- recursive **302**, 313
- shallow **77**
- int
 - zero initialization **56**
- Internet resources **499**
- introspection 436
- intrusive 238
- invasive 238
 - counter 388
- IsArrayT<> 351
- IsClassT<> 266, 276, 359
- IsEnumT<> 356
- IsFuncT<> 352
- IsFundat<> 347
- ISO **G511**
- IsPtrMemT<> 351
- IsPtrT<> 350
- IsRefT<> 350
- iterated instantiation **157**
- iterator 241, **G511**
- iterator traits
 - for pointers 262
- iterator_traits 262, 284

J

- Java 227
- JosuttisAutoPtr 502
- JosuttisOOP 502
- JosuttisStdLib 502

K

- Koenig, Andrew 122, 140
- Koenig lookup 122, 140
- KoenigMooAcc 502

L

- LambdaLib 502
- layout
 - control 224
 - principles 290
- lazy instantiation **143**

- length
 - of symbols 79
- less **429**
- levels of instantiation 313
- lexing **127**
- limit
 - levels of instantiation 313
- linkable entity 90, 155, **G512**
- linkage **99**
 - to C 435
- linking 153
- LippmanObjMod 502
- list 111, 241
- list parameters **222**
- literature **499**
- local class 101
- lookup
 - argument-dependent 122
 - for names 121
 - Koenig lookup 122, 140
 - ordinary 122, 146
 - qualified 120
 - two-phase **146**
 - unqualified 120
- loop
 - split 327
 - unrolling 314
- lvalue 423, **G512**

M

- macro 417
- make_pair() 59
- mangled name 59, 418
- manual instantiation 160
- map 428
- match 488
 - best 487
 - perfect 489
- max_element() 241
- maximum
 - levels of instantiation 313
 - munch 129
- member
 - as base class 293
 - class template **45**, 95
 - function
 - see member function

- function template **45, 95**
- initialization 56
- template
 - see member template
- member class template **G512**
- member function 97
 - and virtual **98**
 - as template **45, 95**
 - implementation 24
 - pointer to 423
 - template 87, **G512**
- member template **45, 95, G512**
 - full specialization **197**
 - versus template template parameter 259
- metaprogramming 203, **301**
 - induction variables 309
 - reflection 363
- Metaware 139, 318
- MeyersCounting 502
- MeyersEffective 502
- MeyersMoreEffective 503
- motivation of templates 7
- MTL 318, 503
- MusserWangDynaVeri 503
- Myers, Nathan 258, 284, 299
- MyersTraits 503

N

- Nackman, Lee R. 174
- name 90, **119, 121**
 - class name injection 126
 - dependent 119, 121
 - dependent of templates **132**
 - dependent of types **130**
 - friend name injection 125
 - linkage 421
 - lookup **119, 121**
 - nondependent 121
 - qualified 119, 120
 - two-phase lookup **146**
 - typedef 101
 - unqualified 120
- name()
 - of `std::type_info` **58, 62, 421**
- named template argument 216, **285**
- namespace
 - associated 123

- template 133
 - unnamed 478, 484
- nested class 97
 - as template **45, 95**
- NewMat 503
- newsgroups 499
- NewShorterOED 503
- NIHCL 243
- nondeduced parameter 14
- nondependent base class 135
- nondependent name 121, **G512**
- nonintrusive 238
- noninvasive 238
 - counter 386
- nonreference
 - versus reference 58, 168, 331, 493
- nontemplate
 - overloading 189
- nontype argument
 - for templates **109**
 - functor 432
- nontype parameter 35, **101**
 - restrictions 40
- null pointer 184
- numeric_limits 284

O

- ODR 90, **475, G512**
- on-demand instantiation 141
- one-definition rule 90, **475, G512**
- operator
 - > 145
 - [] 491
 - >> 206
 - typeof 215
- operator-function-id 120
- optimization
 - for empty base class 289
- oracle **84**
- ordering
 - partial 186
 - rules 188
- order of header files 73
- ordinary lookup 122, 146
- overloading 15, **179, 487**
 - class templates 221
 - nonreference versus reference 493

- of function templates **183**
 - partial ordering 186
 - reference versus nonreference 493
 - templates and nontemplates 189
- overload resolution 356, **487**, **G512**
- P**
- pair 395, 397
- parameter 90, **100**, **G513**
 - actual 91
 - ellipsis 107, 358, 489
 - for base class 45, 137
 - for call **13**
 - formal 91
 - list of types **222**
 - nontype 35, **101**
 - of class templates **173**
 - reference versus nonreference 58, 168, 331
 - template template parameter **50**, **102**
 - type **101**
 - versus argument 90
 - void 410
- parameterization clause 95
- parameterized class **G513**
- parameterized function 481, **G513**
- parameterized traits 254
- parameter
 - match 488
- parsing **127**
 - maximum munch 129
 - of angle brackets 128, 205
- partial ordering
 - of overloading 186
- partial specialization 29, 88, *305*, *331*, **G513**
 - additional parameters 351
 - for function templates 213
 - of class templates **200**
- pattern 239
 - CRTP **295**
- Pennello, Tom 139
- perfect match 489, 492
- POD 87, **G513**
- POI **146**, 480, **G513**
- pointer
 - conversions 494
 - conversion to `void*` 494
 - iterator traits 262
 - qualification 350
 - smart **365**
 - to function 421
 - zero initialization **56**
- pointer-to-member
 - conversion to 392
 - function 423
 - qualification 351
- point of instantiation **146**, 480, **G513**
- policy class **255**, **G513**
 - versus traits 258
- policy traits **275**
- polymorphic object 479
- polymorphism **231**, **G514**
 - bounded 238
 - dynamic **231**, 238
 - static **234**, 238
 - unbounded 238
- POOMA 318, 503
- `pop_back()`
 - for vectors 21, 24
- practice **61**
- precompiled header **72**, **G514**
- prelinker 157
- preprocessor
 - guard 479
- primary template 88, **100**, 201, *301*, *305*, **G514**
- prime numbers 318
- promotion 489
 - of types **271**
 - traits **271**
- property
 - static 218
 - traits 275
- `ptrdiff_t` **4**
 - versus `size_t` 491
- `push_back()`
 - for vectors 21
- Q**
- qualification
 - of array types 351
 - of class types 266, 359
 - of enumeration types 356
 - of function types 352
 - of fundamental types 347
 - of pointer-to-member types 351

- of pointer types 350
- of reference types 350
- qualified-id 120
- qualified lookup 120
- qualified name 119, 120, **G514**
- qualifier 268
- queried instantiation **156**

R

- RAII **373**
- read-only parameter types 276
- recursive duo 401
- recursive instantiation **302**, 313
- reference 268
 - qualification 350
 - to function 421
 - to reference **269**
 - versus nonreference 58, 168, 331, 493
- reference counting **379**, **G514**
- reflection 363
- resource acquisition is initialization **373**
- restricted template expansion 174
- run-time analysis oracles 84
- rvalue 102, **G514**

S

- scanning **127**
- semantic transparency 181
- separation model **68**, **149**
- sequence
 - of conversions 494
- set 428
- SFINAE **106**, 266, 353
- shallow instantiation **77**
- signature 184
- SignSelectT<> 463
- sizeof 263
- size_t **4**
 - versus ptrdiff_t 491
- Smalltalk 238, 243
- smart pointer **365**
 - constness 390
 - implicit conversions 390
- source file **G514**
- spaces 516
- specialization **27**, 88, 141, **179**, **G514**

- explicit 88, 130, 137, **190**
- full *301*
- generated 88
- instantiated 88
- partial 88, **200**, 305, 331
- partial for function templates 213
- Spicer, John 203
- spilled inlined function 156
- split loop 327
- square root 305
- stack 264
- Stack<> **21**
- Standard02 504
- Standard98 503
- Standard Template Library 139, 240
 - see STL
- StaticBoundVal<> 459
- static constant
 - versus enumeration 303
- static member 26, 97
- static polymorphism **234**, 238
- static properties 218
- std::allocator 52, 284
- <std> 74
- std.hpp 73
- std::auto_ptr 394
- std::binary_function 284
- std::bitset 44
- std::char_traits 284
- std::complex 12
- <stddef.h> **4**
- std::deque 52
- std::greater 429
- std::iterator_traits 262, 284
- std::less **429**
- std::list 111, 241
- std::make_pair() 59
- std::map 428
- std::max_element() 241
- std::numeric_limits 284
- std::pair 395, 397
- std::ptrdiff_t **4**
- std::set 428
- std::size_t **4**
- std::stack 264
- std::type_info **58**, 62, 421
- std::unary_function 284

- std::valarray 342
 - std::vector 21, 241
 - STL 139, 240, 342
 - string
 - [] 491
 - and reference template parameters 169
 - as template argument 40, 209
 - literal as template parameters 169
 - StroustrupC++PL 504
 - StroustrupDnE 504
 - StroustrupGlossary 504
 - struct
 - definition 480
 - qualification 266, 359
 - versus class 87
 - Sun Microsystems 156
 - surrogate 496
 - SutterExceptional 504
 - SutterMoreExceptional 504
 - symbol
 - length 79
- T**
- tagged union 224
 - Taligent 139
 - taxonomy of names 119
 - template G514
 - argument 90
 - see template argument
 - export 149
 - for namespaces 133
 - friend 117
 - id
 - see template-id
 - inline 72
 - instantiation 88, 141
 - see instantiation
 - member template 45, 95
 - metaprogramming 203, 301
 - motivation 7
 - name 90, 119
 - nontype arguments 109
 - of template 27
 - parameter 9, 13, 90, 100
 - partial specialization 29
 - primary 100, 201
 - specialization 27, 88
 - type arguments 108
 - union 96
 - .template 44, 132
 - future 207
 - >template 45, 132
 - future 207
 - ::template 132
 - future 207
 - template argument 90, 104, G514
 - array 351
 - char* 40, 209
 - class 40
 - conversions 172
 - deduction 167, G515
 - derived class 295
 - double 40, 210
 - explicit 135
 - float 40, 210
 - named 216, 285
 - string 40, 169, 209
 - template class 87
 - see class template
 - template function 87
 - see function template
 - template-id 87, 90, 104, 120, 132, G515
 - template member function 87
 - see member function template
 - template parameter G515
 - array 351
 - decay 102
 - string literal 169
 - template template argument 51, 111, 211
 - template template parameter 50, 102, 260
 - argument matching 51, 111, 211
 - container 112
 - future 211
 - versus member template 259
 - temporaries 322
 - terminology 87, 507
 - this-> 45, 137
 - tokenization 127
 - maximum munch 129
 - to_string()
 - for bitsets 44
 - tracer 79
 - traits 245, 331, G515
 - CSM 279

- fixed **246**
- for promotion **271**
- for value and reference 331
- parameterized 254
- policy traits **275**
- std::iterator_traits 262
- template **G515**
- value traits 250
- versus policy class 258
- transfer capsule 376
- translation unit 90, 150, **475**, **G515**
- transparency 180
- true constant **G515**
- trule **376**
- tuple **395**, **G515**
- Tuple<> 410
- two-phase lookup 137, **146**, **G515**
- two-stage lookup **146**
- type
 - arguments **108**
 - complete 143
 - conversion 45
 - definition **4**, **27**
 - see typedef
 - dependent name **130**
 - of container element 264
 - promotion **271**
 - qualification 278, **347**, **361**
 - read-only parameter 276
- typedef **4**, **26**, **27**
 - name 101
 - template **212**
- type function **263**, 403
- typeid **58**, 62, 421
- type_info **58**, 62, 421
- typename 10, **43**, 101, 130
 - future 206
- typeof **215**
- type parameter 10, **101**
- type safety 239
- TypeT<> 278, 359
- __type_traits 363

U

- UCN 120
- UDC 358
- unary_function 284

- unbounded polymorphism 238
- union
 - definition 480
 - discriminated 224
 - qualification 266, 359
 - tagged 224
 - template **96**
- universal character name 120
- unnamed namespace 478, 484
- unqualified-id 120
- unqualified lookup 120
- unqualified name 120
- unrolling loops 314
- Unruh, Erwin 301, **318**
- UnruhPrimeOrig 504
- UsedFunctorParam<> 438
- USE_EXPORT 71
- user-defined conversion 109, **G515**
- using 10, 133
- using-declaration 139
 - dependent name **133**

V

- valarray 342
- value
 - as parameter 35
 - binders 457
 - functions 263
- value traits 250
- VandevoordeSolutions 504
- variant type 224
- vector 21, 241
- Veldhuizen, Todd 320, 341
- VeldhuizenMeta95 505
- VeldhuizenPapers 505
- virtual
 - function dispatch table 156
 - member function **98**
 - parameterized **298**
- void
 - as parameter type 410
- void*
 - conversion to 494

W

- Web site of the book **5**

whitespace **G516**

Z

zero initialization **56**