Index

, —see comma operator
! —see logical negation operator
!= —see inequality operator
operator 16–5
operator 16–6
% —see modulus operator
%= operator 5–18
&
 —see address-of operator
 —see bitwise AND operator
 reference declarator 8–5
&& —see logical AND operator
&= operator 5–18
()
 —see function call operator
 function declarator 8–8
*
 —see indirection operator
 —see multiplication operator
 pointer declarator 8–4
*= operator 5–18
+
 —see addition operator
 —see unary plus operator
++ —see increment operator
+= operator 5–11, 18
–
 —see subtraction operator
 —see unary minus operator
–– —see decrement operator
–= operator 5–18
–> —see class member access operator
–>* —see pointer to member operator
. —see class member access operator
.* —see pointer to member operator
... —see ellipsis
/ —see division operator
/* */ comment 2–4
// comment 2–4
/= operator 5–18
:
 field declaration 9–10
 label specifier 6–1
::
 —see scope resolution operator
 scope resolution operator 3–5
::*, pointer to member declarator 8–6
<
 —see less than operator
 template and 14–2
<< —see left shift operator
<<= operator 5–18
<= —see less than or equal to operator
= —see assignment operator
== —see equality operator
> —see greater than operator
>= —see greater than or equal operator
>> —see right shift operator
>>= operator 5–18
?: —see conditional expression operator
[]
 —see subscripting operator
 array declarator 8–7
\ —see backslash
^ —see bitwise exclusive OR operator
^= operator 5–18
_
 underscore character 2–4
 underscore in identifier 2–5
{}
 block statement 6–1
 class declaration 9–1
 class definition 9–1
 enum declaration 7–10
 initializer list 8–13
| —see bitwise inclusive OR operator
|= operator 5–18
|| —see logical OR operator
~
 —see destructor
 —see one's complement operator
0
 —see also zero, null
 null character 2–9
 string terminator 2–9

A

abort() 3–8, 15–5
abs 17–194, 200, 206
abstract
 class 10–8
 class, constructor and 10–9
 class, pointer to 10–9
abstract-declarator 8–2
access

adjusting base class member 11–3
ambiguity, member 10–3
and `friend`, class 11–5
and `friend` function 11–5
base class 11–2
base class member 10–1
class member 5–4
control 11–1
control, anonymous `union` 9–10
control default 11–1
control, member function and 12–1
control, overloading resolution and 10–4
declaration 11–3
declaration, overloaded name and 11–4
default assignment operator 12–9
default copy constructor 12–9
example, member name 11–3
member name 11–1
overloading and 13–3
`protected` member 11–6
rules, template 14–12
specifier 11–1/2
specifier and `friend` 11–6
specifier and object layout 11–2
`struct` default member 9–1
`union` default member 9–1
virtual function 11–7
access-specifier 10–1
addition operator 5–15
additive operator 5–15
additive-expression 5–15
address
of bit-field 9–10
of bit-field restriction 9–10
of constructor 12–1
of overloaded function 5–10, 13–7
of qualified name 5–10
address-of operator 5–10
`adjustfield` 17–45
adjusting base class member access 11–3
adjustment
array parameter 8–9
function parameter 8–9
aggregate 8–13
initialization 12–5
alert 2–7
alias 7–15, 17–5, 11
alignment
of bit-field 9–10
of bit-field, implementation dependency 9–10
requirement, implementation dependency 3–11
`<all>` 17–4/5, 10
`<all.ns>` 17–4/5, 10
allocation
function 12–3
implementation dependency 9–4, 11–2
implementation dependency base class 10–2
implementation dependency bit-field 9–10
`new`, storage 5–11
allowing an exception 15–4
altermate definition 17–13
ambiguity
base class member 10–3
class conversion 10–5
declaration type 7–2
declaration versus cast 8–3
declaration versus expression 6–6
detection, overloaded function 13–3
function declaration 8–13
`if-else` 6–2
member access 10–3
parentheses and 5–13
pointer conversion 4–3
pointer to member conversion 4–3
reference conversion 4–3
resolution, scoping 10–4
Amendment 1 17–2, 12, 61
anachronism C–10
C function definition C–10
assignment to `this` C–11
cast of pointer to member C–12
free store and constructor C–11
free store and destructor C–11
memory management C–11
nonnested class C–12
old style base class initializer C–11
old style function definition C–10
`overload` keyword C–10
pointer to member conversion C–12
scope of nested class C–12
`this` and constructor C–11
`this` and destructor C–11
AND
operator, bitwise 5–17
operator, logical 5–17
operator, side effects and logical 5–18
anonymous
`union` 9–9
`union` access control 9–10
`union`, extension to C C–1
`union`, global 9–10
`union` restriction 9–10
`app` 17–45
`arg` 17–194, 200, 206
`argc` 3–7
argument 1–1, 17–12, 15, 22/23, 29/30, 38, 40, 53, 94/96, 120, 163/164, 170/171, 180/183
and name hiding, default 8–11
binding of default 8–10
class object as 12–5
conversion 5–4, 8–9
declaration, default 8–10
deduction, template 14–14
evaluation of default 8–10/11
evaluation, order of 5–4
evaluation, unspecified order of 5–4
example of default 8–10
list, empty 8–8
list, variable 8–8
matching —see overloading resolution
overloaded operator and default 13–9
overloading and default 8–11
passing 5–4
passing, reference and 8–15
reference 5–4
scope of default 8–11
specification, template 14–14
substitution 16–5
template 14–12
temporary and default 12–1
to constructor, unspecified 5–13
type checking 5–4
type checking of default 8–10
type conversion 12–1
type, unknown 8–8
`argv[]` 3–7
arithmetic
conversion 4–2
exception 5–1
exception, implementation dependency 5–1
extension to C single precision C–1
pointer 5–15

single precision floating point 4−1
type 3−11
unsigned 3−11
array
bound 8−7
const 7−7
constructor and 5−12
declaration 8−7
declarator [] 8−7
declarator, multidimensional 8−7
default constructor and 5−12
example 8−7
initialization 8−13
member 9−4
multidimensional 8−8
new 5−12
of class objects and constructor 12−5
of class objects and default constructor 12−5
of class objects and new 5−12
of class objects initialization 8−14, 12−5
order of execution, constructor and 12−1
order of execution, destructor and 12−2
overloading and pointer versus 13−2
parameter adjustment 8−9
pointer conversion 4−3
size, default 8−7
sizeof 5−11
storage of 8−8
type 3−11, 8−9
arrow operator —see class member access operator
asm
declaration 7−20
implementation dependency 7−20
assembler 7−20
<assert.h> 17−4
assignment
and initialization, overloaded 12−5
and lvalue 5−18
base class object 5−19
const pointer 5−19
conversion by 5−19
derived class object 5−19
expression 5−18
extension to C memberwise C−2
member 12−8
memberwise 13−9
of class object 12−9
of derived class to base class 12−9
operator 5−18, 12−8, 17−15, 43, 179
operator access, default 12−9
operator, default 13−9
operator, default 12−8/9
operator, overloaded 13−9
operator restriction, default 12−8/9
pointer to const 5−19
pointer to member 5−19
pointer to volatile 5−19
reference 8−15
to class object 5−19
to pointer 5−19
to pointer to member 5−19
to pointer to member, zero 5−19
to pointer, zero 5−19
to reference 5−19
to this anachronism C−11
volatile pointer 5−19
assignment-expression 5−18
assignment-operator 5−19
associated sequence 17−65, 121/122, 126/127
ate 17−45
atexit 17−22
atexit() 3−8
auto
destruction of 6−4/5
initialization 6−6
object initialization 8−12
restriction 7−3
specifier 7−2
storage duration 3−9
automatic initialization 6−5/6

B

backslash character 2−7
backspace 2−7
badbit 17−45
badcast::badcast 17−29
badcast::~badcast 17−29
badtypeid::badtypeid 17−42
badtypeid::~badtypeid 17−42
badyypeid::do_raise 17−42
base
class 17−13, 15, 18, 25, 27/33, 36, 42, 47/49, 64, 76, 80, 90, 99, 105/107, 109, 113/114, 116, 121, 123/124, 126/127, 187/188
class 10−1/2
class access 11−2
class allocation, implementation dependency 10−2
class, assignment of derived class to 12−9
class cast 5−7
class constructor order of execution 12−1
class destructor order of execution 12−1
class, direct 10−1
class, indirect 10−1
class initialization 12−6
class initialization, order of 12−6
class initializer 8−12
class initializer anachronism, old style C−11
class member access 10−1
class member access, adjusting 11−3
class member ambiguity 10−3
class object, assignment 5−19
class pointer conversion 4−3
class, private 11−2
class, public 11−2
class, reference to 4−3
class virtual —see virtual base class
of integer literal 2−6
basefield 17−45
base-specifier 10−1
base-specifier-list 10−1
beg 17−45
behavior
default 17−14, 27, 34, 38, 40/41, 73, 75/77, 105, 113
implementation-defined 1−2
locale-specific 1−2
required 17−14
undefined 1−2
unspecified 1−2
Ben 13−2
binary
mode 17−50
operator, interpretation of 13−9
operator, overloaded 13−9
binary 17−45
binding
—see virtual function, dynamic
of default argument 8−10
bit-field 9−10
address of 9−10
alignment of 9−10

allocation, implementation dependency 9–10
declaration 9–10
implementation dependency alignment of 9–10
implementation dependency sign of 9–10
layout 9–10
restriction 9–10
restriction, address of 9–10
restriction, pointer to 9–10
type of 9–10
unnamed 9–10
zero width of 9–10
bit-fields, Boolean 3–11
bitmask type 17–18, 47, 49/50, 98
<bits> 17–4, 162
bits<N>::any 17–167
bits<N>::bits 17–164
bits<N>::count 17–166
bits<N>::length 17–166
bits<N>::none 17–167
bits<N>::operator!= 17–167
bits<N>::operator&= 17–164
bits<N>::operator<< 17–167
bits<N>::operator<<= 17–165
bits<N>::operator== 17–167
bits<N>::operator>> 17–167
bits<N>::operator>>= 17–165
bits<N>::operator^= 17–165
bits<N>::operator|= 17–164
bits<N>::operator~ 17–166
bits<N>::reset 17–165
<bits.ns> 17–4
bits<N>::set 17–165
bits<N>::test 17–167
bits<N>::toggle 17–166
bits<N>::to_string 17–166
bits<N>::to_ulong 17–166
bits<N>::to_ushort 17–166
<bitstring> 17–4, 168
bitstring::any 17–176
bitstring::append 17–172
bitstring::assign 17–172
bitstring::bitstring 17–170/171
bitstring::count 17–175
bitstring::find 17–175
bitstring::insert 17–173
bitstring::length 17–175
bitstring::none 17–176
<bitstring.ns> 17–4
bitstring::operator!= 17–176
bitstring::operator&= 17–171
bitstring::operator+= 17–171
bitstring::operator<< 17–177
bitstring::operator<<= 17–172
bitstring::operator== 17–176
bitstring::operator>> 17–177
bitstring::operator>>= 17–172
bitstring::operator^= 17–172
bitstring::operator|= 17–171
bitstring::operator~ 17–174
bitstring::remove 17–173
bitstring::replace 17–173
bitstring::reset 17–174
bitstring::resize 17–175
bitstring::rfind 17–176
bitstring::set 17–174
bitstring::substr 17–176
bitstring::test 17–176
bitstring::toggle 17–174
bitstring::to_string 17–174
bitstring::trim 17–175
bitwise
AND operator 5–17
exclusive OR operator 5–17
inclusive OR operator 5–17
operator 5–17
block
initialization in 6–5
scope —see local scope
statement {} 6–1
structure 6–5
body, function 8–11
bool
increment 5–5, 11
integer conversion 4–1
type-specifier 7–8
Boolean
bit-fields 3–11
constant 2–9
conversion 4–4
literal 2–9
type 3–10
type 3–11
boolean-literal 2–9
bound array 8–7
bound, of array 8–7
bound pointer to member function, undefined C–12
break statement 6–4/5
buffer, stream 17–47/48, 51, 64, 77/79, 87/89, 97, 107, 115, 124, 128/129
buffered file 17–22, 123/128
built-in type —see fundamental type
byte 5–11
string, null-terminated 17–19

C

C
anonymous union, extension to C–1
class, extension to C–1
const, extension to C–1
dangerous extension to C–10
declaration statement, extension to C–1
delete, extension to C–1
destructor, extension to C–2
expression evaluation, difference from C–1
extension to C–1/2
function definition anachronism C–10
header 17–4/5, 10, 12, 16, 20, 23
headers, ISO 2–5
implementation dependency extension to C–10
inline function, extension to C–1
library, Standard 17–2, 4, 12, 20, 40
linkage to 7–20
memberwise assignment, extension to C–2
memberwise initialization, extension to C–2
multiple inheritance, extension to C–2
new, extension to C–1
overloading delete, extension to C–2
overloading, extension to C–1
overloading new, extension to C–2
pointer to member, extension to C–2
protected, extension to C–2
reference type, extension to C–1
single precision arithmetic, extension to C–1
stream 17–22, 51, 80, 128
summary, compatibility with C–1
summary, compatibility with ISO C–2
type checking, extension to C–1
user-defined type, extension to C–1
void* pointer type extension to C–1
volatile, extension to C–2

call
—see also function call, member function call, overloaded function call, virtual function call
by reference 5–4
by value 5–4
operator function 13–8
calloc 17–22, 38
capacity 17–23
carriage return 2–7
case label 6–1, 3
<cassert.ns> 17–4
cast
ambiguity, declaration versus 8–3
base class 5–7
class object 5–8
const 5–10
derived class 5–7
dynamic 5–5, 17–28
implementation dependency pointer to function 5–9
integer to pointer 5–9
lvalue 5–8/9
of pointer to member anachronism C–12
operator 5–10, 14, 8–2
pointer to function 5–9
pointer to integer 5–9
pointer to member 5–7, 9
reference 5–7, 9
reinterpret 5–9
reinterpretcast, lvalue 5–9
cast, reinterpretcast, reference 5–9
cast
static 5–7
staticcast, class object 5–8
staticcast, lvalue 5–8
cast, staticcast, reference 5–7
cast to incomplete class 5–9
cast-expression 5–14
casting 5–4, 14
catch 15–1
c-char 2–7
c-char-sequence 2–7
<cctype.ns> 17–4
<cerrno.ns> 17–4
<cfloat.ns> 17–4
C++
header 17–4/5, 10
library, Standard 17–2, 4, 12/13, 18/20, 23, 25, 29/31, 38, 40, 48, 61, 129
change to string literal, undefined 2–9
char
implementation dependency sign of 3–10
integer conversion 4–1
literal, implementation dependency value of 2–8
type 3–10
type, signed 3–10/11
type specifier 7–8
type, unsigned 3–10/11
character
array initialization 8–15
constant 2–7
decimal-point 17–19, 49
literal 2–7
literal, type of 2–7
multibyte 1–2
signed 3–10
string 2–9
type 3–10
underscore 17–11
character-literal 2–7
checking
point of error 14–3
syntax 14–3
<ciso646.ns> 17–4
class 3–11, 9–1
abstract 10–8
access and friend 11–5
anachronism, nonnested C–12
and type 9–1
base 17–13, 15, 18, 25, 27/33, 36, 42, 47/49, 64, 76, 80, 90, 99, 105/107, 109, 113/114, 116, 121, 123/124, 126/127, 187/188
base —see base class
cast to incomplete 5–9
constructor and abstract 10–9
conversion 12–1
conversion ambiguity 10–5
declaration, forward 9–2, 10–1
declaration {} 9–1
definition 9–1, 3
definition 3–3
definition example 9–4
definition name hiding 9–2
definition, scope of 9–2
definition {} 9–1
derived 17–18
derived —see derived class
extension to C C–1
friend 11–5
generated 14–7
lattice —see DAG
linkage of 3–6
linkage specification 7–20
local —see local class
member —see also member
member access 5–4
member access operator 5–4
member declaration 9–3
member function 9–6
member initialization 8–13
member semantics 5–4
member, static 3–9
member storage duration 3–9
member syntax 5–4
name 8–2
name as type definition 9–1
name declaration 3–2
name, elaborated 7–9, 9–2
name, point of declaration 9–3
name, scope of 9–2
name, typedef 7–6, 9–3
nested —see nested class
object as argument 12–5
object, assignment of 12–9
object, assignment to 5–19
object cast 5–8
object cast, staticcast, 5–8
object, const 7–7, 9–7
object copy 12–8
object copy —see also copy constructor
object copy example 12–9
object initialization 8–13, 12–5
object initialization —see also constructor
object layout 9–4, 10–2
object, member 9–4
object, operations on 9–1
object return type 12–5
object, sizeof 5–11
objects and constructor, array of 12–5
objects and default constructor, array of 12–5
objects and new, array of 5–12
objects initialization, array of 8–14, 12–5
pointer to abstract 10–9

polymorphic 10−6
scope 3−5
scope of enumerator 7−11
`sizeof`, empty 9−1
specialized 14−7
template 14−2, 17−94/96, 162/163, 178/179, 187
type restriction, member of 12−6
unnamed 7−6
`class`
type specifier 7−9
versus `struct` 9−1
versus `union` 9−1
class-key 7−9, 9−1
class-name 9−1
class-specifier 9−1
`<climits.ns>` 17−4
`<clocale.ns>` 17−4
`<cmath.ns>` 17−4
comma
operator 5−19
operator, side effects and 5−19
comment 2−2
`/* */` 2−4
`//` 2−4
comparison
implementation dependency pointer 5−17
pointer 5−16/17
pointer to function 5−16
undefined pointer 5−16/17
`void*` pointer 5−16
compatibility
with C summary C−1
with ISO C summary C−2
compilation, separate 2−1
compiler control line —see preprocessing directive
complete object 1−3
completely-defined object type 3−10
`<complex>` 17−4, 191
`<complex.ns>` 17−4
compound
statement 6−1
type 3−11
compound-statement 6−1
concatenation
string 2−9
undefined string literal 2−9
condition conversion 4−4
condition 6−2
conditional
expression operator 5−18
inclusion 16−2
conditional-expression, throw-expression in 5−18
*condition*s, rules for 6−2
`conj` 17−195, 200, 206
consistency
example, linkage 7−3
linkage 3−7, 7−3
linkage specification 7−20
type declaration 3−7
const cast 5−10
`*const` example 8−4
`const` 3−12
array 7−7
assignment, pointer to 5−19
class object 7−7, 9−7
constructor and 9−8, 12−1
destructor and 9−8, 12−1
example 8−4
extension to C C−1
initialization 7−7, 8−12
initialization, pointer to 8−12
linkage of 3−6, 7−3
member function 9−7
member initialization 12−6
operand 5−1
overloading and 13−1/2
pointer assignment 5−19
pointer initialization 8−12
reference 8−15
type 7−6
`void*` pointer conversion 4−2
`volatile void*` pointer conversion 4−2
constant 2−6, 3−11, 5−2
character 2−7
enumeration 7−10
expression 5−20
expression, pointer to member 5−10
integer 2−6
`long` 2−6
pointer declaration 8−4
pointer example 8−4
`unsigned` 2−6
constant-expression 5−20
construction, order of 3−9
constructor 12−1
access, default copy 12−9
address of 12−1
anachronism, free store and C−11
anachronism, `this` and C−11
and abstract class 10−9
and array 5−12
and array order of execution 12−1
and `const` 9−8, 12−1
and initialization 12−5
and initialization example 12−5
and member function 12−1
and member function call 12−7
and `new` 5−12
and `new`, implementation dependency 5−13
and `return` 6−5
and `static` objects order of execution 12−6
and virtual function call 12−7
and `volatile` 9−8, 12−1
array of class objects and 12−5
call, explicit 12−1
conversion by 12−1
conversion by —see also user-defined conversion
copy 12−1, 8, 17−15, 43, 179
default 17−68, 179/180, 185
default —see default constructor
default copy 12−8/9
definition 8−12
example 12−1
exception handling 15−2
for temporary 12−1
inheritance of 12−1
local object 3−9
order of execution, base class 12−1
order of execution, member 12−1
restriction 12−1
restriction, default copy 12−8/9
type of 12−1
`union` 9−9
unspecified argument to 5−13
`continue`
in `for` statement 6−4
statement 6−4/5
control line —see preprocessing directive
conversion
—see also type conversion
Boolean 4−4
ambiguity, class 10−5

ambiguity, pointer 4–3
ambiguity, pointer to member 4–3
ambiguity, reference 4–3
anachronism, pointer to member C–12
and name hiding, user-defined 12–1
argument 5–4, 8–9
arithmetic 4–2
array pointer 4–3
base class pointer 4–3
`bool` integer 4–1
by assignment 5–19
by constructor 12–1
`char` integer 4–1
class 12–1
condition 4–4
`const void*` pointer 4–2
`const volatile void*` pointer 4–2
derived class pointer 4–3
explicit type —see casting
floating point integer 4–2
function —see also user-defined conversion
implementation defined pointer integer 5–9
implementation dependency floating point 4–1
implementation dependency integer 4–1
implicit 4–1, 5–1, 12–1
implicit user-defined 12–1
inheritance of user-defined 12–1
integer 4–1
lvalue 4–1
null pointer 4–3
operator 5–1, 12–1
out of range value, undefined 4–1
overloaded function and standard 13–6
overloading resolution and 13–5
overloading resolution and pointer 13–8
overloading resolution and standard 13–5
overloading resolution and user-defined 13–6
pointer 4–2
pointer to function 4–3
pointer to member 4–3
pointer to member `void*` 4–4
reference 4–3
return type 6–5
rules, type 4–2
safe floating point 4–1
`signed unsigned` integer 4–1
standard 4–1
to enumeration type 5–7
to enumeration type, `staticcast`, 5–7
type of 12–1
user-defined 5–1, 12–1
virtual user-defined 12–1
`void*` pointer 4–2
`volatile const void*` pointer 4–2
`volatile void*` pointer 4–2
zero pointer 4–3
conversion-function-id 12–1
conversions
cv-qualifier pointer 4–2
cv-qualifier reference 4–3
copy
class object 12–8
constructor 12–1, 8, 17–15, 43, 179
constructor access, default 12–9
constructor, default 12–8/9
constructor, implicitly-declared 12–1
constructor restriction, default 12–8/9
example, class object 12–9
`cos` 17–195, 200, 206
`cosh` 17–195, 200, 206
`__cplusplus` 16–9
`<csetjmp.ns>` 17–4
`<csignal.ns>` 17–4
`<cstdarg.ns>` 17–4
`<cstddef.ns>` 17–4
`<cstdio.ns>` 17–4
`<cstdlib.ns>` 17–4, 20
`<cstring.ns>` 17–4
`<ctime.ns>` 17–4
ctor-initializer 12–6
`<ctype.h>` 17–4, 80, 82, 148
`cur` 17–45
cv-qualifier 3–12
pointer conversions 4–2
reference conversions 4–3
cv-qualifier 8–2
`<cwchar.ns>` 17–4
`<cwctype.ns>` 17–4

D

DAG
multiple inheritance 10–3
nonvirtual base class 10–3
virtual base class 10–3
dangerous extension to C C–10
data member —see member
deallocation —see `delete`
`dec` 17–45, 57, 81, 91
decimal literal 2–6
decimal-literal 2–6
decimal-point character 17–19, 49
declaration 3–1/2, 7–1
`:`, field 9–10
access 11–3
ambiguity, function 8–13
array 8–7
as definition 7–2
`asm` 7–20
bit-field 9–10
class member 9–3
class name 3–2
class name, point of 9–3
consistency, type 3–7
constant pointer 8–4
default argument 8–10
definition versus 3–2
ellipsis in function 5–4, 8–8
enumerator, point of 3–6
example 3–2, 8–9
example, function 8–10
`extern` 3–2
`extern`, point of 3–6
`extern` reference 8–15
forward 7–4
forward class 9–2, 10–1
`friend`, point of 3–6
function 3–2, 8–8
function member 9–6
function template 14–16
hiding —see name hiding
in `for`, scope of 6–4
in `for` statement 6–4
in `switch` statement 6–3
matching, overloaded function 13–2
member 9–3
multiple 3–7
name 3–2
name, point of 3–6
overloaded name and access 11–4
overloaded name and `friend` 11–5

parameter 8–8/9
parentheses in 8–3/4
pointer 8–4
reference 8–5
`register` 7–3
specifier 7–2
statement 6–5
statement, extension to C C–1
`static` member 3–2
storage class 7–2
type 8–4
type ambiguity 7–2
`typedef` 3–2
`typedef` as type 7–5
versus cast ambiguity 8–3
versus expression ambiguity 6–6
`{}`, class 9–1
`{}`, `enum` 7–10
declaration 7–1
declaration-seq 7–20
declaration-statement 6–5
declarative region 3–1
declarator 7–1, 8–1
`&`, reference 8–5
`()`, function 8–8
`*`, pointer 8–4
`::*`, pointer to member 8–6
`[]`, array 8–7
example 8–2
initializer, temporary and 12–1
meaning of 8–4
multidimensional array 8–7
declarator 8–1
declarator-id 8–2
decl-specifier 7–2
decrement
operator 5–5, 10/11
operator, overloaded 13–10
deduction, template argument 14–14
default
access control 11–1
argument and name hiding 8–11
argument, binding of 8–10
argument declaration 8–10
argument, evaluation of 8–10/11
argument, example of 8–10
argument, overloaded operator and 13–9
argument, overloading and 8–11
argument, overloading resolution and 13–4
argument, scope of 8–11
argument, temporary and 12–1
argument, type checking of 8–10
array size 8–7
assignment operator 13–9
assignment operator 12–8/9
assignment operator access 12–9
assignment operator restriction 12–8/9
behavior 17–14, 27, 34, 38, 40/41, 73, 75/77, 105, 113
constructor 17–68, 179/180, 185
constructor 12–1, 6, 8
constructor and array 5–12
constructor and initialization 12–5
constructor and `new` 5–12
constructor, array of class objects and 12–5
copy constructor 12–8/9
copy constructor access 12–9
copy constructor restriction 12–8/9
destructor 12–1
initialization 8–13
member access, `struct` 9–1
member access, `union` 9–1
template parameter 14–10
`default` label 6–1, 3
`default_size` 17–23
`#define` 16–5
`<defines>` 17–4, 23
`<defines.ns>` 17–4
definition 3–2, 17–19
altermate 17–13
and initialization 7–2
class 3–3
class 9–1, 3
class name as type 9–1
constructor 8–12
declaration as 7–2
enumerator 3–3
enumerator point of 7–11
example 3–2
example, function 8–12
example, nested class 9–11
function 3–3
function 8–11
function template 14–16
inline member function 3–7
local class 9–12
member 9–7
member function 9–6/8
name hiding, class 9–2
namespace 7–12
nested class 9–10
object 3–3
of template 14–1
pure virtual function 10–8
scope, macro 16–6
scope of class 9–2
scope of function 3–7
`static` member 9–9
versus declaration 3–2
virtual function 10–7
`{}`, class 9–1
definitions, implementation-generated 3–2
`delete` 5–13/14, 12–3
destructor and 5–14, 12–2
example 12–4
example, destructor and 12–4
example, scope of 12–4
extension to C C–1
extension to C overloading C–2
`operator` 17–13, 38, 40
overloading and 12–4
type of 12–4
undefined 5–13
undefined value 5–13
`delete[]`, `operator` 17–13, 38, 40
deleted object, undefined 5–13
delete-expression 5–13
dependent name 14–5
deprecated features 5–5, 11
dereferencing 5–1
—see also indirection
derivation —see inheritance
derived
class 17–18
class 10–1
class cast 5–7
class example 10–1
class, most 12–6
class object, assignment 5–19
class, overloading and 13–2
class pointer conversion 4–3
class to base class, assignment of 12–9
destruction

of `auto` 6–4/5
of local `static` 6–6
of local variable 6–4/5
of temporary 12–1
of temporary, order of 12–1
order of 3–9
destructor 12–1, 17–15, 34, 128, 179
anachronism, free store and C–11
anachronism, `this` and C–11
and array order of execution 12–2
and `const` 9–8, 12–1
and `delete` 5–14, 12–2
and `delete` example 12–4
and exit from scope 6–4
and fundamental type 12–3
and member function 12–2
and member function call 12–7
and placement of object 12–2
and `static` objects order of execution 12–6
and virtual function call 12–7
and `volatile` 9–8, 12–1
call example, explicit 12–2
call, explicit 12–2
call, implicit 12–2
call, unspecified 6–6
default 12–1
exception handling 15–2
extension to C C–2
for temporary 12–1
inheritance of 12–1
local object 3–9
order of execution 12–1
order of execution, base class 12–1
order of execution, member 12–1
program termination and 12–2
pure virtual 12–2
restriction 12–1/2
`static` object 3–8
`union` 9–9
virtual 12–2
diagnostic message 1–1
difference from C expression evaluation C–1
digit 2–4
digit-sequence 2–8
digraph 2–3, 5
direct base class 10–1
direct-abstract-declarator 8–2
direct-declarator 8–1
directed acyclic graph —see DAG
directive
error 16–8
null 16–9
pragma 16–8
preprocessing 16–1
direct-new-declarator 5–11
distinct string 2–9
division
by zero, undefined 5–1, 15
implementation dependency 5–15
operator 5–14
`do` statement 6–3/4
dominance, virtual base class 10–5
dot operator —see class member access operator
double quote 2–7
`double`
literal 2–8
type 3–11
type specifier 7–8
`_double_complex` 17–197, 202/205
`double_complex::double_complex` 17–197
dynamic
binding —see virtual function
cast 5–5, 17–28
initialization 3–8
type 1–1
`<dynarray>` 17–4, 178
`<dynarray.ns>` 17–4
`dynarray<T>` 17–178/181, 185/188

E

`E` suffix 2–8
elaborated
class name 7–9, 9–2
`enum` name 7–9
type specifier 14–12
type specifier 3–5
type specifier —see elaborated class name
elaborated-type-specifier 7–9
`#elif` 16–2
elimination of temporary 12–1
ellipsis
example 8–10
in function declaration 5–4, 8–8
overloading resolution and 13–4/6
`#else` 16–3
`else` 6–2
empty
argument list 8–8
class `sizeof` 9–1
statement 6–1
`end` 17–45
`#endif` 16–3
`endl` 17–91, 94
end-of-file 17–50, 59, 81, 83/86, 93, 148, 168, 177
`ends` 17–94
entity 3–1
enum name, `typedef` 7–6
`enum`
declaration `{}` 7–10
name, elaborated 7–9
overloading and 13–1
type of 7–10/11
type specifier 7–9
enumerated type 3–10, 17–17/18, 23, 47, 51
enumeration 7–10
constant 7–10
example 7–11
type, conversion to 5–7
type, `staticcast`, conversion to 5–7
underlying type 7–11
enumerator
class, scope of 7–11
definition 3–3
linkage of 3–6
member 7–12
point of declaration 3–6
point of definition 7–11
redefinition 7–11
restriction 7–11
value of 7–10
enumerator 7–10
environment, program 3–7
`EOF` 17–59
`eofbit` 17–45
equality operator 5–17
equality-expression 5–17
equivalence
template type 14–13
type 7–5, 9–1
`<errno.h>` 17–4, 11

error
checking, point of 14–3
directive 16–8
#error 16–8
escape
character —see backslash
sequence 2–7
sequence, undefined 2–8
escape-sequence 2–7
evaluation
difference from C expression C–1
new, unspecified order of 5–13
of default argument 8–10/11
of expression, order of 5–1
order of argument 5–4
unspecified order of 5–1
unspecified order of argument 5–4
unspecified order of function call 5–4
exact match, overloading resolution 13–5
example
*const 8–4
array 8–7
class definition 9–4
class object copy 12–9
const 8–4
constant pointer 8–4
constructor 12–1
constructor and initialization 12–5
declaration 3–2, 8–9
declarator 8–2
definition 3–2
delete 12–4
derived class 10–1
destructor and delete 12–4
ellipsis 8–10
enumeration 7–11
explicit destructor call 12–2
explicit qualification 10–4
friend 9–2
friend function 11–5
function declaration 8–10
function definition 8–12
linkage consistency 7–3
local class 9–12
member function 9–6, 11–5
member name access 11–3
nested class 9–10
nested class definition 9–11
nested class forward declaration 9–11
nested type name 9–12
of default argument 8–10
of incomplete type 3–10
overloading 13–1
pointer to member 8–6
pure virtual function 10–9
scope of delete 12–4
scope resolution operator 10–4
static member 9–8
subscripting 8–7
type name 8–2
typedef 7–5
unnamed parameter 8–12
variable parameter list 8–10
virtual function 10–6/7
exception
allowing an 15–4
arithmetic 5–1
declaration scope 3–4
handler 15–3, 17–13, 34
handling 15–1
handling constructor 15–2
handling destructor 15–2
implementation dependency arithmetic 5–1
throwing 15–1
<exception> 17–4, 24, 40
exception-declaration 15–1
<exception.ns> 17–4
exception-specification 15–4
exit from scope, destructor and 6–4
exit 17–22, 34, 40
exit() 3–7/8
EXIT_FAILURE 17–22
EXIT_SUCCESS 17–22
exp 17–195, 200, 206
explanation, subscripting 8–7
explicit
constructor call 12–1
destructor call 12–2
destructor call example 12–2
instantiation syntax 14–9
qualification 3–5, 7–19
qualification example 10–4
type conversion —see casting
exponent-part 2–8
expression 5–1
ambiguity, declaration versus 6–6
assignment 5–18
constant 5–20
evaluation, difference from C C–1
order of evaluation of 5–1
parenthesized 5–2
pointer to member constant 5–10
postfix 5–3
primary 5–2
reference 5–1
statement 6–1
unary 5–10
unspecified 5–4
expression 5–19
expression-list 5–3
expression-statement 6–1
extension
to C C–1/2
to C anonymous union C–1
to C class C–1
to C const C–1
to C, dangerous C–10
to C declaration statement C–1
to C delete C–1
to C destructor C–2
to C, implementation dependency C–10
to C inline function C–1
to C memberwise assignment C–2
to C memberwise initialization C–2
to C multiple inheritance C–2
to C new C–1
to C overloading C–1
to C overloading delete C–2
to C overloading new C–2
to C pointer to member C–2
to C protected C–2
to C reference type C–1
to C single precision arithmetic C–1
to C type checking C–1
to C user-defined type C–1
to C, void* pointer type C–1
to C volatile C–2
extern
"C" 17–11/12
"C++" 17–11/12
declaration 3–2
linkage of 7–3

linkage specification 7−20
point of declaration 3−6
reference declaration 8−15
restriction 7−3
external linkage 3−6, 17−10/12

F

`F` suffix 2−8
`f` suffix 2−8
`failbit` 17−45
`fclose` 17−117
`fflush` 17−121
`fgetc` 17−119, 124
field declaration `:` 9−10
File scope 3−4
file 2−1
buffered 17−22, 123/128
open 17−50
scope 17−11
seek 17−50
source 2−1, 17−2, 4
unbuffered 17−128
`filebuf::close` 17−117
`filebuf::filebuf` 17−116
`filebuf::~filebuf` 17−116
`filebuf::is_open` 17−116
`filebuf::open` 17−116/117
`filebuf::overflow` 17−117, 125
`filebuf::pbackfail` 17−118, 125
`filebuf::seekoff` 17−120, 126
`filebuf::seekpos` 17−120, 126
`filebuf::sync` 17−121, 126
`filebuf::uflow` 17−119, 125
`filebuf::underflow` 17−118, 125
`filebuf::xsgetn` 17−119
`filebuf::xsputn` 17−119
final overrider 10−6
floctal-digit 2−6
`fixed` 17−45, 57
`float`
literal 2−8
type 3−11
type specifier 7−8
`_float_complex` 17−192
`float_complex::float_complex` 17−191
`floatfield` 17−45
`<float.h>` 2−5, 17−4
floating
point arithmetic, single precision 4−1
point conversion, implementation dependency 4−1
point conversion, safe 4−1
point integer conversion 4−2
point literal 2−8
point literal, type of 2−8
point type 3−10
point type 3−11
point type, implementation dependency 3−11
floating-constant 2−8
floating-suffix 2−8
`flush` 17−52, 80, 88, 91, 94, 121
`fopen` 17−116/117
`for`
scope of declaration in 6−4
statement 6−3/4
statement, `continue` in 6−4
statement, declaration in 6−4
form feed 2−7
formal
argument —see also parameter
argument —see parameter
forward
class declaration 9−2, 10−1
declaration 7−4
declaration example, nested class 9−11
`fpos_t` 17−60/61
`fprintf` 17−89
`fputc` 17−117/118, 124
fractional-constant 2−8
free
store —see also `new`, `delete`
store and constructor anachronism C−11
store and destructor anachronism C−11
friend
function, scope of 11−6
specifier 17−19
`friend`
access specifier and 11−6
class 11−5
class access and 11−5
declaration, overloaded name and 11−5
example 9−2
function, access and 11−5
function example 11−5
function, inline 11−6
function, linkage of 11−6
function, member function and 11−5
function, nested class 9−12
inheritance and 11−6
member function 11−5
point of declaration 3−6
specifier 7−6
template and 14−18
`virtual` and 10−7
`fscanf` 17−79
`fseek` 17−117, 120
`fsetpos` 17−120
`<fstream>` 17−4, 16, 115
`<fstream.ns>` 17−4
function
—see also `friend` function, member function, inline function, virtual function
allocation 12−3
argument —see argument
body 8−11
call 5−4
call evaluation, unspecified order of 5−4
call operator 5−3, 13−8
call operator, overloaded 13−9
call, recursive 5−4
call, undefined 5−7, 9
cast, implementation dependency pointer to 5−9
cast, pointer to 5−9
comparison, pointer to 5−16
conversion, pointer to 4−3
declaration 3−2, 8−8
declaration ambiguity 8−13
declaration, ellipsis in 5−4, 8−8
declaration example 8−10
declaration matching, overloaded 13−2
declarator `()` 8−8
definition 8−11
definition 3−3
definition anachronism, C C−10
definition anachronism, old style C−10
definition example 8−12
definition, scope of 3−7
generated 14−7
global 17−11/12, 15
handler 17−13
linkage specification 7−20

linkage specification overloaded 7–20
member —see member function
member declaration 9–6
name hiding 13–2
name, overloaded 13–1
`operator` 13–8
overloaded —see also overloading
parameter —see parameter
parameter adjustment 8–9
pointer to member 5–14
prototype scope 3–4
return —see `return`
return type —see return type
scope 3–4
scope of friend 11–6
specialized 14–7
specifier 7–4
template 14–13
template declaration 14–16
template definition 14–16
type 3–11, 8–8/9
virtual —see virtual function
virtual member 17–13, 15, 76
function-body 8–11
function-definition 8–11
function-like macro 16–4
function-specifier 7–4
fundamental
type 3–10
type conversion —see conversion, user-defined conversion
type, destructor and 12–3
`fvoid_t` 17–23

G

generated
class 14–7
constructor —see default constructor
destructor —see default destructor
function 14–7
`getline` 17–77, 79, 85/86, 148
global
anonymous `union` 9–10
function 17–11/12, 15
name 3–4
scope 3–4
`goodbit` 17–45
`goto`
initialization and 6–5
statement 6–1, 4/5
grammar A–1
greater
than operator 5–16
than or equal to operator 5–16

H

handler
exception 15–3, 17–13, 34
function 17–13
handler 15–1
handler-seq 15–1
handling exception —see exception handling
header
C 17–4/5, 10, 12, 16, 20, 23
C++ 17–4/5, 10
primary 17–4, 10
secondary 17–4, 10, 12
headers
ISO C 2–5
library 2–5
standard 2–5
hex number 2–8
`hex` 17–45, 57
hexadecimal literal 2–6
hexadecimal-digit 2–6
hexadecimal-escape-sequence 2–7
hexadecimal-literal 2–6
hiding —see name hiding
horizontal tab 2–7

I

id, qualified 5–2
identifier 2–4, 5–2, 7–1
_, underscore in 2–5
identifier 2–4
identities and overloading, operator 13–9
id-expression 5–2
id-expression 5–2
`#if` 16–2, 17–12
`if` statement 6–2
`#ifdef` 16–3
`if-else` ambiguity 6–2
`#ifndef` 16–3
`ifstream::close` 17–122
`ifstream::ifstream` 17–121
`ifstream::~ifstream` 17–122
`ifstream::is_open` 17–122
`ifstream::open` 17–122
`ifstream::rdbuf` 17–122
`imag` 17–192/195, 197, 199/200, 202/203, 205/206
`imanip<T>` 17–95
`imanip<T>::imanip` 17–95
implementation
defined pointer integer conversion 5–9
dependency `__STDC__` 16–9
dependency alignment of bit-field 9–10
dependency alignment requirement 3–11
dependency allocation 9–4, 11–2
dependency arithmetic exception 5–1
dependency `asm` 7–20
dependency base class allocation 10–2
dependency bit-field allocation 9–10
dependency constructor and `new` 5–13
dependency division 5–15
dependency extension to C C–10
dependency floating point conversion 4–1
dependency floating point type 3–11
dependency generation of temporary 12–1
dependency integer conversion 4–1
dependency left shift 5–16
dependency linkage of `main()` 3–7
dependency linkage specification 7–20
dependency modulus 5–15
dependency object linkage 7–21
dependency overflow 5–1
dependency parameters to `main()` 3–7
dependency pointer comparison 5–17
dependency pointer subtraction 5–16
dependency pointer to function cast 5–9
dependency promotion of `wchar_t` 4–1
dependency range of types 2–5
dependency sign of bit-field 9–10
dependency sign of `char` 3–10
dependency `signed` `unsigned` 4–1
dependency `sizeof` expression 5–11
dependency `sizeof` integral type 3–11
dependency `sizeof` type 3–10

dependency string literal 2–9
dependency type of integer literal 2–7
dependency type of `ptrdiff_t` 5–16
dependency type of `size_t` 5–11
dependency type of `sizeof` expression 2–5
dependency value of `char` literal 2–8
dependency value of multicharacter literal 2–7
dependency `volatile` 7–7
dependency `wchar_t` 3–11
limits 1–2
type 17–17
implementation-defined 17–4, 14, 20, 22/23, 40, 53
behavior 1–2
implementation-generated definitions 3–2
implicit
conversion 4–1, 5–1, 12–1
destructor call 12–2
user-defined conversion 12–1
implicitly-declared
copy constructor 12–1
default constructor 12–1
`in` 17–45
`#include` 16–3, 17–2, 5
inclusion
conditional 16–2
source file 16–3
incomplete
class, cast to 5–9
type 3–10
type, example of 3–10
incompletely-defined object type 3–10
increment
`bool` 5–5, 11
operator 5–5, 10/11
operator, overloaded 13–10
indeterminate uninitialized variable 8–13
indirect base class 10–1
indirection 5–10
operator 5–10
inequality operator 5–17
inheritance 10–1
—see also multiple inheritance
and `friend` 11–6
of constructor 12–1
of destructor 12–1
of overloaded operator 13–9
of user-defined conversion 12–1
init-declarator 8–1
init-declarator-list 8–1
initialization 8–12
aggregate 12–5
and `goto` 6–5
and `new` 5–12, 12–5
array 8–13
array of class objects 8–14, 12–5
`auto` 6–6
`auto` object 8–12
automatic 6–5/6
base class 12–6
character array 8–15
class member 8–13
class object 8–13, 12–5
class object —see also constructor
`const` 7–7, 8–12
`const` member 12–6
`const` pointer 8–12
constructor and 12–5
default 8–13
default constructor and 12–5
definition and 7–2
dynamic 3–8
example, constructor and 12–5
extension to C memberwise C–2
in block 6–5
jump past 6–3, 5
local object 3–9
local `static` 6–6
member 12–6, 8
member object 12–6
non-trivial 12–5
order of 3–8/9, 10–2
order of base class 12–6
order of member 12–6
order of virtual base class 12–6
overloaded assignment and 12–5
parameter 5–4
pointer to `const` 8–12
pointer to `volatile` 8–12
reference 8–6, 15
reference member 12–6
run-time 3–8
`static` member 9–9
`static` object 3–8, 8–12/13
`struct` 8–13
`union` 8–14, 9–9
virtual base class 12–6, 8
`volatile` pointer 8–12
initializer 8–12
base class 8–12
list `{}` 8–13
member 8–12
scope of member 12–7
temporary and declarator 12–1
initializer 8–12
initializer-clause 8–12
initializer-list 8–12
injection from template, name 14–6
inline 17–12
`friend` function 11–6
function 7–4
function, extension to C C–1
function, linkage of 7–3/4
member function 7–4, 9–8
member function definition 3–7
member function rewriting rules 9–8
`inline`
linkage of 3–6
specifier 7–4
instantiation
and specialization 14–8
multiple 14–8
point of 14–7
syntax, explicit 14–9
template 14–7
`int`
type 3–11
type specifier 7–8
type, `unsigned` 3–11
integer
cast, pointer to 5–9
constant 2–6
conversion 4–1
conversion, `bool` 4–1
conversion, `char` 4–1
conversion, floating point 4–2
conversion, implementation defined pointer 5–9
conversion, implementation dependency 4–1
conversion, `signed` `unsigned` 4–1
literal 2–6
literal, base of 2–6
literal, implementation dependency type of 2–7
literal, type of 2–7

to pointer cast 5–9
type 3–11
integer-literal 2–6
integer-suffix 2–6
integral
promotion 4–1, 5–4
type 3–10
type 3–11
type, implementation dependency `sizeof` 3–11
value, undefined unrepresentable 4–2
internal linkage 3–6
`internal` 17–45, 57
interpretation
of binary operator 13–9
of unary operator 13–9
`invalidargument::do_raise` 17–29
`invalidargument::invalidargument` 17–29
`invalidargument::~invalidargument` 17–29
invocation, macro 16–5
`<iomanip>` 17–4, 94
`<iomanip.ns>` 17–4
`<ios>` 17–4, 10, 45
`ios::bad` 17–54
`ios::clear` 17–53
`ios::copyfmt` 17–52
`ios::eof` 17–54
`ios::exceptions` 17–54
`ios::fail` 17–54
`ios::failure::do_raise` 17–49
`ios::failure::failure` 17–49
`ios::failure::~failure` 17–49
`ios::fill` 17–55
`ios::flags` 17–55
`ios::fmtflags` 17–49, 96/97
`ios::good` 17–54
`ios::init` 17–57, 80, 90
`ios::Init::Init` 17–51
`ios::Init::~Init` 17–52
`ios::ios` 17–52, 56
`ios::~ios` 17–52
`ios::iostate` 17–50
`ios::io_state` 17–51
`ios::iword` 17–56
`<ios.ns>` 17–4, 10
`ios::openmode` 17–50, 63, 66, 76, 97, 104, 108/109, 112/117, 120, 123, 126
`ios::open_mode` 17–51, 63, 66, 115, 117
`ios::operator void*` 17–52
`ios::operator!` 17–52
`ios::precision` 17–55
`ios::pword` 17–56
`ios::rdbuf` 17–53
`ios::rdstate` 17–53
`ios::seekdir` 17–51, 63, 66, 76, 97, 104, 108, 112, 115, 120, 123, 126
`ios::seek_dir` 17–51, 63, 66
`ios::setf` 17–55
`ios::setstate` 17–53/54
`ios::tie` 17–52/53
`<iostream>` 17–4, 51, 128
`<iostream.ns>` 17–4
`ios::unsetf` 17–55
`ios::width` 17–56
`ios::xalloc` 17–56
ISO
C headers 2–5
C summary, compatibility with C–2
`<iso646.h>` 17–4, 20
`iso_standard_library` 17–10/11, 20
`isspace` 17–80/82, 148
`istdiostream::buffered` 17–127
`istdiostream::istdiostream` 17–126
`istdiostream::~istdiostream` 17–126
`istdiostream::rdbuf` 17–127
`<istream>` 17–4, 77
`istream::gcount` 17–87
`istream::get` 17–84/85
`istream::getline` 17–85/86
`istream::ignore` 17–86
`istream::ipfx` 17–80
`istream::isfx` 17–81
`istream::istream` 17–80
`istream::~istream` 17–80
`<istream.ns>` 17–4
`istream::operator>>` 17–81/83
`istream::peek` 17–87
`istream::putback` 17–87
`istream::read` 17–86/87
`istream::sync` 17–87
`istream::unget` 17–87
`istringstream::istringstream` 17–113
`istringstream::~istringstream` 17–113
`istringstream::rdbuf` 17–114
`istringstream::str` 17–114
`istrstream::istrstream` 17–105/106
`istrstream::~istrstream` 17–106
`istrstream::rdbuf` 17–106
iteration statement 6–3
iteration-statement 6–3, 5
scope 6–3

J

Jessie 12–1
jump
past initialization 6–3, 5
statement 6–4
jump-statement 6–4

K

keyword A–1
anachronism, `overload` C–10
list 2–4
mutable 3–9
Koenig, Andrew 15–1

L

`L`
prefix 2–7, 9
suffix 2–7/8
`l` suffix 2–7/8
label 6–5
`case` 6–1, 3
`default` 6–1, 3
name space 6–1
scope of 3–4, 6–1
specifier : 6–1
labeled statement 6–1
lattice —see DAG, class
layout
access specifier and object 11–2
bit-field 9–10
class object 9–4, 10–2
left
shift, implementation dependency 5–16
shift operator 5–16
shift, undefined 5–16
`left` 17–45, 57

length of name 2–4
lengtherror::do_raise 17–30
lengtherror::lengtherror 17–30
lengtherror::~lengtherror 17–30
less
 than operator 5–16
 than or equal to operator 5–16
lexical conventions 2–1
library
 Standard C 17–2, 4, 12, 20, 40
 Standard C++ 17–2, 4, 12/13, 18/20, 23, 25, 29/31, 38, 40, 48, 61, 129
 headers 2–5
limits, implementation 1–2
<limits.h> 2–5, 3–10, 17–4, 81, 86, 100, 148, 164
#line 16–8
linkage 3–1, 6
 consistency 3–7, 7–3
 consistency example 7–3
 external 3–6, 17–10/12
 implementation dependency object 7–21
 internal 3–6
 of class 3–6
 of const 3–6, 7–3
 of enumerator 3–6
 of extern 7–3
 of friend function 11–6
 of inline 3–6
 of inline function 7–3/4
 of local name 3–7
 of main(), implementation dependency 3–7
 of member function 3–7
 of static 3–6, 7–3, 21
 of static member 3–7
 of template 3–6
 of typedef 3–6
 specification 7–20
 specification class 7–20
 specification consistency 7–20
 specification, extern 7–20
 specification function 7–20
 specification, implementation dependency 7–20
 specification object 7–20/21
 specification overloaded function 7–20
 to C 7–20
linkage-specification 7–20
list
 keyword 2–4
 operator 2–5, 13–8
 {}, initializer 8–13
literal 2–6, 5–2
 base of integer 2–6
 character 2–7
 decimal 2–6
 double 2–8
 float 2–8
 floating point 2–8
 hexadecimal 2–6
 implementation dependency string 2–9
 implementation dependency type of integer 2–7
 implementation dependency value of char 2–8
 implementation dependency value of multicharacter 2–7
 integer 2–6
 long 2–6/7
 long double 2–8
 multicharacter 2–7
 octal 2–6
 type of character 2–7
 type of floating point 2–8
 type of integer 2–7
 unsigned 2–6/7
literal 2–6
local
 class definition 9–12
 class example 9–12
 class member function 9–12
 class, member function in 9–8
 class restriction 9–12
 class restriction, static member 9–8
 class, scope of 9–12
 name, linkage of 3–7
 object constructor 3–9
 object destructor 3–9
 object initialization 3–9
 object, static 3–9
 object storage duration 3–9
 scope 3–4
 static, destruction of 6–6
 static initialization 6–6
 variable, destruction of 6–4/5
<locale.h> 17–4, 19/20
locale-specific behavior 1–2
log 17–195, 201, 206
logical
 AND operator 5–17
 AND operator, side effects and 5–18
 OR operator 5–18
 OR operator, side effects and 5–18
 negation operator 5–10/11
long
 constant 2–6
 double literal 2–8
 double type 3–11
 literal 2–6/7
 type 3–11
 type specifier 7–8
 type, unsigned 3–11
 typedef and 7–2
long_double_complex 17–202/203, 207
long_double_complex::long_double_complex 17–202
longjmp 17–22
long-suffix 2–6
lookup
 name 3–1
 template name 14–2
lowercase 17–19, 49
lvalue 3–13
 assignment and 5–18
 cast 5–8/9
 cast, reinterpretcast, 5–9
 cast, staticcast, 5–8
 conversion 4–1
 modifiable 3–13

M

macro
 definition scope 16–6
 function-like 16–4
 invocation 16–5
 masking 17–12
 name 16–5
 object-like 16–4
 parameters 16–5
 preprocessor 16–1
 replacement 16–4
main() 3–7
 implementation dependency linkage of 3–7
 implementation dependency parameters to 3–7

parameters to 3–7
`return` from 3–7/8
`malloc` 17–22, 38
management anachronism, memory C–11
masking macro 17–12
`<math.h>` 17–4
meaning of declarator 8–4
member
—see also base class member
access operator, overloaded 13–10
access ambiguity 10–3
access, base class 10–1
access, class 5–4
access, `protected` 11–6
access, `struct` default 9–1
access, `union` default 9–1
array 9–4
assignment 12–8
cast, pointer to 5–7, 9
class object 9–4
constructor order of execution 12–1
declaration 9–3
declaration, class 9–3
declaration, `static` 3–2
definition 9–7
definition, `static` 9–9
destructor order of execution 12–1
enumerator 7–12
example, `static` 9–8
function and access control 12–1
function and `friend` function 11–5
function call, constructor and 12–7
function call, destructor and 12–7
function call, undefined 9–7
function, class 9–6
function, `const` 9–7
function, constructor and 12–1
function definition 9–6/8
function definition, inline 3–7
function, destructor and 12–2
function example 9–6, 11–5
function, `friend` 11–5
function in local class 9–8
function in nested class 9–8
function, inline 7–4, 9–8
function, linkage of 3–7
function, local class 9–12
function, nested class 9–11
function, overloading resolution and 13–4
function rewriting rules, inline 9–8
function, `static` 9–6, 8
function template 14–17
function, `union` 9–9
function, virtual 17–13, 15, 76
function, `volatile` 9–7/8
initialization 12–6, 8
initialization, `const` 12–6
initialization, order of 12–6
initialization, reference 12–6
initialization, `static` 9–9
initializer 8–12
initializer, scope of 12–7
linkage of `static` 3–7
local class restriction, `static` 9–8
name access 11–1
name access example 11–3
name, overloaded 9–3
object initialization 12–6
of class type restriction 12–6
pointer to —see pointer to member
`static` 9–8
`static` class 3–9
storage duration, class 3–9
template and `static` 14–18
type of `static` 5–10, 9–9
use, `static` 9–8
member-declaration 9–3
member-declarator 9–3
member-specification 9–3
memberwise
assignment 13–9
assignment, extension to C C–2
initialization, extension to C C–2
`memchr` 17–20
`memcmp` 17–145, 160
mem-initializer 12–6
memory
management —see also `new`, `delete`
management anachronism C–11
model 1–3
message, diagnostic 1–1
missing storage class specifier 7–3
mode
binary 17–50
text 17–50
modifiable lvalue 3–13
modulus
implementation dependency 5–15
operator 5–14
most derived class 12–6
multibyte
character 1–2
string, null-terminated 17–19
multicharacter
literal 2–7
literal, implementation dependency value of 2–7
multidimensional
array 8–8
array declarator 8–7
multiple
declaration 3–7
inheritance 10–1/2
inheritance DAG 10–3
inheritance, extension to C C–2
inheritance, `virtual` and 10–7
instantiation 14–8
multiplication operator 5–14
multiplicative operator 5–14
multiplicative-expression 5–14
mutable keyword 3–9

N

name 2–4, 3–1, 5–2
address of qualified 5–10
and translation unit 3–1
class —see class name
declaration 3–2
dependent 14–5
elaborated `enum` 7–9
global 3–4
hiding 3–5/6, 5–2, 6–5
hiding, class definition 9–2
hiding, function 13–2
hiding, overloading versus 13–2
hiding, user-defined conversion and 12–1
injection from template 14–6
length of 2–4
linkage of local 3–7
lookup 3–1
lookup, template 14–2

macro 16–5
overloaded function 13–1
overloaded member 9–3
point of declaration 3–6
qualified 7–19
reserved 17–11
resolution, template 14–2
scope of 3–4
space, label 6–1
type —see type name
unreserved 17–16
use of template 14–4
namespace 17–5, 10/11, 20
definition 7–12
scope 3–4
namespaces 7–12
NDEBUG 17–4
negation operator, logical 5–10/11
nested
class anachronism, scope of C–12
class definition 9–10
class definition example 9–11
class example 9–10
class forward declaration example 9–11
class friend function 9–12
class member function 9–11
class, member function in 9–8
class, scope of 9–10
type name 9–12
type name example 9–12
type name, scope of 9–12
<new> 17–4, 10, 13, 22, 35
new 5–11/12, 12–3
array 5–12
array of class objects and 5–12
constructor and 5–12
default constructor and 5–12
extension to C C–1
extension to C overloading C–2
implementation dependency constructor and 5–13
initialization and 5–12, 12–5
operator 17–13, 22, 37/38, 40/41
placement syntax 5–12
scoping and 5–12
storage allocation 5–11
type of 12–3
unspecified order of evaluation 5–13
unspecified value 5–12
new[], operator 17–13, 37/38, 40/41
new-declarator 5–11
new-expression 5–11
<new.h> 2–5
new-initializer 5–12
new-line 2–7
<new.ns> 17–4
new-placement 5–11
new-type-id 5–11
nondigit 2–4
nonnested class anachronism C–12
non-trivial
implicitly-declared default constructor 12–1
initialization 12–5
nonvirtual base class DAG 10–3
nonzero-digit 2–6
norm 17–195, 201, 207
noshowbase 17–58
noshowpoint 17–58
noskipws 17–58
notation, syntax 1–2
nouppercase 17–58
NPOS 17–23
NTBS 17–19, 26, 89, 91, 94, 105/107, 116
static 17–19
NTMBS 17–19, 25/27, 29/31, 36, 42, 49
NTWCS 17–20
static 17–20
null
character 0 2–9
directive 16–9
pointer 4–2/3, 5–16
pointer conversion 4–3
reference 8–6
statement 6–1
null-terminated
byte string 17–19
multibyte string 17–19
wide-character string 17–20
number
hex 2–8
octal 2–8

O

object 1–3, 3–1/2, 13
class —see also class object
complete 1–3
constructor, local 3–9
definition 3–3
destructor and placement of 12–2
destructor, local 3–9
destructor static 3–8
initialization, auto 8–12
initialization, local 3–9
initialization, static 3–8, 8–12/13
layout, access specifier and 11–2
linkage, implementation dependency 7–21
linkage specification 7–20/21
static local 3–9
storage duration, local 3–9
temporary —see temporary
type, completely-defined 3–10
type, incompletely-defined 3–10
undefined deleted 5–13
unnamed 12–1
object-like macro 16–4
oct 17–45, 58
octal
literal 2–6
number 2–8
octal-escape-sequence 2–7
octal-literal 2–6
offsetof 17–22
ofstream 17–122/123
ofstream::close 17–123
ofstream::is_open 17–123
ofstream::ofstream 17–123
ofstream::~ofstream 17–123
ofstream::open 17–123
ofstream::rdbuf 17–123
old
style base class initializer anachronism C–11
style function definition anachronism C–10
omanip<T> 17–95/96
omanip<T>::omanip 17–96
one-definition rule 3–3
one's complement operator 5–10/11
open file 17–50
operand
const 5–1
reference 5–1
volatile 5–1

operations on class object 9–1
operator
 %= 5–18
 &= 5–18
 *= 5–18
 += 5–11, 18
 -= 5–18
 /= 5–18
 <<= 5–18
 >>= 5–18
 ^= 5–18
 additive 5–15
 address-of 5–10
 assignment 5–18, 12–8, 17–15, 43, 179
 bitwise 5–17
 bitwise AND 5–17
 bitwise exclusive OR 5–17
 bitwise inclusive OR 5–17
 cast 5–10, 14, 8–2
 class member access 5–4
 comma 5–19
 conditional expression 5–18
 conversion 5–1, 12–1
 decrement 5–5, 10/11
 default assignment 13–9
 delete —see delete
 division 5–14
 equality 5–17
 example, scope resolution 10–4
 function call 5–3, 13–8
 function call 13–8
 greater than 5–16
 greater than or equal to 5–16
 identities and overloading 13–9
 increment 5–5, 10/11
 indirection 5–10
 inequality 5–17
 left shift —see left shift operator
 less than 5–16
 less than or equal to 5–16
 list 2–5, 13–8
 logical AND 5–17
 logical OR 5–18
 logical negation 5–10/11
 modulus 5–14
 multiplication 5–14
 multiplicative 5–14
 new —see new
 one's complement 5–10/11
 overloaded 5–1
 overloading —see also overloaded operator
 overloading restrictions 13–8
 pointer to member 5–14
 precedence of 5–1
 relational 5–16
 right shift; right shift operator 5–16
 scope resolution 3–5, 5–2, 9–6, 10–1, 8
 shift —see left shift operator, right shift operator
 side effects and comma 5–19
 side effects and logical AND 5–18
 side effects and logical OR 5–18
 sizeof 5–10/11
 subscripting 5–3, 13–8
 unary 5–10
 unary minus 5–10
 unary plus 5–10
 use, scope resolution 9–8
 |= 5–18
operator
 delete 17–13, 38, 40
 delete[] 17–13, 38, 40
 function 13–8
 new 17–13, 22, 37/38, 40/41
 new[] 17–13, 37/38, 40/41
 overloaded 13–8
operator!= 17–42/43, 60, 63, 147, 161/162, 167/168, 176, 194, 199, 205
operator& 17–18, 167, 177
operator&= 17–18, 162, 164, 168, 171
operator* 17–193, 198, 204
operator*= 17–191/192, 196/197, 202/203
operator+ 17–60, 62, 146, 160/161, 177, 186, 190/193, 198/199, 203, 205
operator+= 17–60, 62, 130, 134/135, 149, 153, 168, 171, 178, 182, 186, 188, 191, 196/197, 202/203
operator- 17–60, 62, 192/193, 198/199, 203/205
operator-= 17–60, 62, 191, 196/197, 202/203
operator/ 17–193, 198/199, 204
operator/= 17–191/192, 196/197, 202/203
operator<< 17–57, 88, 91/93, 95/96, 148, 162, 167/168, 177/178, 194, 200, 206
operator== 17–42/43, 60, 62, 147, 161/162, 167/168, 176, 193/194, 199, 205
operator>> 17–77/78, 81/83, 93, 95, 148, 162, 167/168, 177, 194, 200, 205
operator^ 17–18, 168, 177
operator^= 17–18, 162, 165, 168, 172
operator| 17–18, 167, 177
operator~ 17–18, 162, 166, 168, 174
operator 13–8
operator-function-id 13–8
operators in expressions, overloaded 13–11
optimization of temporary —see elimination of temporary
OR
 operator, bitwise exclusive 5–17
 operator, bitwise inclusive 5–17
 operator, logical 5–18
 operator, side effects and logical 5–18
order
 of argument evaluation 5–4
 of argument evaluation, unspecified 5–4
 of base class initialization 12–6
 of construction 3–9
 of destruction 3–9
 of destruction of temporary 12–1
 of evaluation new, unspecified 5–13
 of evaluation of expression 5–1
 of evaluation, unspecified 5–1
 of execution, base class constructor 12–1
 of execution, base class destructor 12–1
 of execution, constructor and array 12–1
 of execution, constructor and static objects 12–6
 of execution, destructor 12–1
 of execution, destructor and array 12–2
 of execution, destructor and static objects 12–6
 of execution, member constructor 12–1
 of execution, member destructor 12–1
 of function call evaluation, unspecified 5–4
 of initialization 3–8/9, 10–2
 of member initialization 12–6
 of virtual base class initialization 12–6
ostdiostream::buffered 17–128
ostdiostream::ostdiostream 17–127
ostdiostream::~ostdiostream 17–127
ostdiostream::rdbuf 17–127
<ostream> 17–4, 87
ostream::flush 17–94
<ostream.ns> 17–4
ostream::operator<< 17–91/93
ostream::opfx 17–91
ostream::osfx 17–91

ostream::ostream 17–90
ostream::~ostream 17–90
ostream::put 17–93
ostream::write 17–93/94
ostringstream::ostringstream 17–114
ostringstream::~ostringstream 17–115
ostringstream::rdbuf 17–115
ostringstream::str 17–115
ostrstream::freeze 17–107
ostrstream::ostrstream 17–106/107
ostrstream::~ostrstream 17–107
ostrstream::pcount 17–107
ostrstream::rdbuf 17–107
ostrstream::str 17–107
out of range value, undefined conversion 4–1
out 17–45
outofrange 17–30/31, 132, 151, 163, 170, 180
outofrange::do_raise 17–31
outofrange::outofrange 17–30
outofrange::~outofrange 17–30
overflow 5–1
 implementation dependency 5–1
overflow::do_raise 17–31
overflow::overflow 17–31
overflow::~overflow 17–31
overload resolution, template 14–16
overload keyword anachronism C–10
overloaded
 assignment and initialization 12–5
 assignment operator 13–9
 binary operator 13–9
 decrement operator 13–10
 function, address of 5–10, 13–7
 function ambiguity detection 13–3
 function and standard conversion 13–6
 function call operator 13–9
 function call resolution —see also argument matching, overloading resolution
 function declaration matching 13–2
 function, linkage specification 7–20
 function name 13–1
 increment operator 13–10
 member access operator 13–10
 member name 9–3
 name and access declaration 11–4
 name and friend declaration 11–5
 operator 13–8
 operator 5–1
 operator 13–8
 operator and default argument 13–9
 operator, inheritance of 13–9
 operators in expressions 13–11
 subscripting operator 13–10
 unary operator 13–9
overloading 8–8, 9–2, 13–1
 and access 13–3
 and const 13–1/2
 and default argument 8–11
 and delete 12–4
 and derived class 13–2
 and enum 13–1
 and pointer 13–1
 and pointer versus array 13–2
 and reference 13–1
 and return type 13–1
 and scope 13–2
 and specialization 14–17
 and static 13–1
 and typedef 13–1
 and volatile 13–1/2
 delete, extension to C C–2
 example 13–1
 extension to C C–1
 new, extension to C C–2
 operator identities and 13–9
 postfix ++ and -- 13–10
 prefix ++ and -- 13–10
 resolution 13–3
 resolution and access control 10–4
 resolution and conversion 13–5
 resolution and default argument 13–4
 resolution and ellipsis 13–4/6
 resolution and member function 13–4
 resolution and pointer conversion 13–8
 resolution and promotion 13–5
 resolution and standard conversion 13–5
 resolution and user-defined conversion 13–6
 resolution exact match 13–5
 resolution rules 13–5
 resolution, template function 14–14
 resolution trivial conversions 13–5
 restriction 13–9
 subsequence rule 13–5
 versus name hiding 13–2
overrider, final 10–6

P

parameter 1–2, 8–12
 adjustment, array 8–9
 adjustment, function 8–9
 declaration 8–8/9
 default template 14–10
 example, unnamed 8–12
 initialization 5–4
 list example, variable 8–10
 list, variable 5–4, 8–8
 reference 8–5
 scope of 3–4
 void 8–8
parameter type list 8–9
parameter-declaration 8–8
parameterized type —see template
parameters
 macro 16–5
 to main() 3–7
 to main(), implementation dependency 3–7
parentheses
 and ambiguity 5–13
 in declaration 8–3/4
parenthesized expression 5–2
period 17–19
phases, translation 2–1
placement
 of object, destructor and 12–2
 syntax, new 5–12
pm-expression 5–14
POD-struct 8–15
point
 of declaration class name 9–3
 of declaration enumerator 3–6
 of declaration extern 3–6
 of declaration friend 3–6
 of declaration name 3–6
 of definition, enumerator 7–11
 of error checking 14–3
 of instantiation 14–7
 type, floating 3–10
pointer
 —see also void*
 arithmetic 5–15

assignment, const 5–19
assignment to 5–19
assignment, volatile 5–19
cast, integer to 5–9
comparison 5–16/17
comparison, implementation dependency 5–17
comparison, undefined 5–16/17
comparison, void* 5–16
conversion 4–2
conversion ambiguity 4–3
conversion, array 4–3
conversion, base class 4–3
conversion, const void* 4–2
conversion, const volatile void* 4–2
conversion, derived class 4–3
conversion, null 4–3
conversion, overloading resolution and 13–8
conversion, void* 4–2
conversion, volatile const void* 4–2
conversion, volatile void* 4–2
conversion, zero 4–3
conversions, cv-qualifier 4–2
declaration 8–4
declarator * 8–4
example, constant 8–4
initialization, const 8–12
initialization, volatile 8–12
integer conversion, implementation defined 5–9
null 4–2/3, 5–16
overloading and 13–1
subtraction, implementation dependency 5–16
terminology 3–12
to abstract class 10–9
to bit-field restriction 9–10
to const assignment 5–19
to const initialization 8–12
to function cast 5–9
to function cast, implementation dependency 5–9
to function comparison 5–16
to function conversion 4–3
to integer cast 5–9
to member 3–12, 5–14
to member anachronism, cast of C–12
to member, assignment 5–19
to member, assignment to 5–19
to member cast 5–7, 9
to member constant expression 5–10
to member conversion 4–3
to member conversion ambiguity 4–3
to member conversion anachronism C–12
to member declarator ::* 8–6
to member example 8–6
to member, extension to C C–2
to member function 5–14
to member function, undefined bound C–12
to member operator 5–14
to member void* conversion 4–4
to member, zero assignment to 5–19
to volatile assignment 5–19
to volatile initialization 8–12
type 3–11
type extension to C, void* C–1
versus array, overloading and 13–2
zero 4–2/3, 5–16
zero assignment to 5–19
polar 17–195, 201, 207
polymorphic
class 10–6
type 10–6
position, stream 17–61/62, 65, 75/76, 103/105, 112, 119/120
postfix
++ and -- 5–5
++ and --, overloading 13–10
expression 5–3
potential scope 3–1
pow 17–195/196, 201, 207
pragma directive 16–8
#pragma 16–8
precedence of operator 5–1
prefix
++ and -- 5–11
++ and --, overloading 13–10
L 2–7, 9
preprocessing 16–1
directive 16–1
preprocessing-token 2–2
preprocessor, macro 16–1
primary
expression 5–2
header 17–4, 10
private 11–1
base class 11–2
program 3–6
environment 3–7
start 3–7/8
startup 17–2, 13
termination 3–7/8
termination and destructor 12–2
promotion
integral 4–1, 5–4
of wchar_t, implementation dependency 4–1
overloading resolution and 13–5
protected 11–1
extension to C C–2
member access 11–6
protection 17–19
—see access control
ptrdiff_t 5–16, 17–23
implementation dependency type of 5–16
<ptrdynarray> 17–4, 186
<ptrdynarray.ns> 17–4
ptrdynarray<T> 17–179, 186/188
ptrdynarray<T>::append 17–188
ptrdynarray<T>::assign 17–188
ptrdynarray<T>::base 17–190
ptrdynarray<T>::get_at 17–189
ptrdynarray<T>::insert 17–189
ptrdynarray<T>::length 17–190
ptrdynarray<T>::operator+= 17–188
ptrdynarray<T>::operator[] 17–190
ptrdynarray<T>::ptrdynarray 17–187/188
ptrdynarray<T>::put_at 17–189
ptrdynarray<T>::remove 17–189
ptrdynarray<T>::reserve 17–190
ptrdynarray<T>::resize 17–190
ptrdynarray<T>::sub_array 17–189
ptrdynarray<T>::swap 17–189
ptr-operator 8–1
public 11–1
base class 11–2
punctuators 2–5
pure
specifier 9–3
virtual destructor 12–2
virtual function 10–8/9
virtual function call, undefined 10–9, 12–8
virtual function definition 10–8
virtual function example 10–9
pure-specifier 9–3

Q

qualification, explicit 3–5, 7–19
qualified
 id 5–2
 name 7–19
 name, address of 5–10
qualified-id 5–2
question mark 2–7
quote
 double 2–7
 single 2–7

R

`raise` 17–13, 25, 37, 40, 53, 132, 151, 163, 170, 180
range of types, implementation dependency 2–5
reach 3–5
`real` 17–192/194, 196/197, 199/203, 205/207
`realloc` 17–22, 38
recursive function call 5–4
redefinition
 enumerator 7–11
 `typedef` 7–5
reference 3–11
 and argument passing 8–15
 and `return` 8–15
 and temporary 8–15
 argument 5–4
 assignment 8–15
 assignment to 5–19
 call by 5–4
 cast 5–7, 9
 cast, `reinterpretcast`, 5–9
 cast, `staticcast`, 5–7
 `const` 8–15
 conversion 4–3
 conversion ambiguity 4–3
 conversions, cv-qualifier 4–3
 declaration 8–5
 declaration, `extern` 8–15
 declarator `&` 8–5
 expression 5–1
 initialization 8–6, 15
 member initialization 12–6
 null 8–6
 operand 5–1
 overloading and 13–1
 parameter 8–5
 restriction 8–6
 `sizeof` 5–11
 temporary, scope of 8–16
 to base class 4–3
 type, extension to C C–1
 `volatile` 8–16
references 3–9
region, declarative 3–1
`register`
 declaration 7–3
 restriction 7–3
reinterpret cast 5–9
`reinterpretcast`
 lvalue cast 5–9
 reference cast 5–9
relational operator 5–16
relational-expression 5–16
remainder operator —see modulus operator
replacement, macro 16–4
required behavior 17–14
reraise 15–2
rescanning and replacement 16–6
`reserve` 17–23
reserved
 identifier 2–5
 name 17–11
 word —see keyword
`resetiosflags` 17–96
resolution
 overloading —see overloading resolution
 scoping ambiguity 10–4
 template function overloading 14–14
 template name 14–2
 template overload 14–16
restriction 17–12/13
 address of bit-field 9–10
 anonymous `union` 9–10
 `auto` 7–3
 bit-field 9–10
 constructor 12–1
 default assignment operator 12–8/9
 default copy constructor 12–8/9
 destructor 12–1/2
 enumerator 7–11
 `extern` 7–3
 local class 9–12
 member of class type 12–6
 overloading 13–9
 pointer to bit-field 9–10
 reference 8–6
 `register` 7–3
 `static` 7–3
 `static` member local class 9–8
 `union` 9–9, 12–1
restrictions, operator overloading 13–8
rethrow 15–2
return
 type 8–9
 type, class object 12–5
 type conversion 6–5
 type, overloading and 13–1
`return` 6–4/5
 constructor and 6–5
 from `main()` 3–7/8
 reference and 8–15
 statement —see also `return`
rewriting rules, inline member function 9–8
right shift operator 5–16
`right` 17–45, 58
rounding 4–2
rule, one-definition 3–3
rules
 for *condition*s 6–2
 inline member function rewriting 9–8
 overloading resolution 13–5
 summary, scope 10–9
 type conversion 4–2
run-time initialization 3–8
rvalue 3–13

S

safe floating point conversion 4–1
s-char 2–9
s-char-sequence 2–9
`scientific` 17–45, 58
scope 3–1
 File 3–4
 class 3–5
 destructor and exit from 6–4
 exception declaration 3–4
 file 17–11

function 3–4
function prototype 3–4
global 3–4
iteration-statement 6–3
local 3–4
macro definition 16–6
namespace 3–4
of class definition 9–2
of class name 9–2
of declaration in `for` 6–4
of default argument 8–11
of `delete` example 12–4
of enumerator class 7–11
of friend function 11–6
of function definition 3–7
of label 3–4, 6–1
of local class 9–12
of member initializer 12–7
of name 3–4
of nested class 9–10
of nested class anachronism C–12
of nested type name 9–12
of parameter 3–4
of reference temporary 8–16
overloading and 13–2
potential 3–1
resolution operator 3–5, 5–2, 9–6, 10–1, 8
resolution operator `::` 3–5
resolution operator example 10–4
resolution operator use 9–8
rules summary 10–9
selection-statement 6–2
scoping
ambiguity resolution 10–4
and `new` 5–12
secondary header 17–4, 10, 12
seek file 17–50
selection statement 6–2
selection-statement 6–2
scope 6–2
semantics, class member 5–4
separate
compilation 2–1
translation 2–1
sequence
associated 17–65, 121/122, 126/127
statement 6–1
sequencing operator —see comma operator
`setbase` 17–97
`setfill` 17–97
`setiosflags` 17–96
`setjmp` 17–11
`<setjmp.h>` 17–4, 11, 22
`setlocale` 17–19
`set_new_handler` 17–37
`setprecision` 17–97
`setterminate()` 15–5
`set_terminate` 17–33
`setunexpected()` 15–5
`set_unexpected` 17–33
`setvbuf` 17–120
`setw` 17–97
shift operator —see left shift operator, right shift operator
shift-expression 5–16
`short`
type 3–11
type specifier 7–8
type, `unsigned` 3–11
`typedef` and 7–2
`showbase` 17–45, 58
`showpoint` 17–45, 58
`showpos` 17–45, 58/59
side
effects 5–1
effects and comma operator 5–19
effects and logical AND operator 5–18
effects and logical OR operator 5–18
sign
of bit-field, implementation dependency 9–10
of `char`, implementation dependency 3–10
sign 2–8
`<signal.h>` 17–4
signature 1–2
`signed`
`char` type 3–10/11
character 3–10
`typedef` and 7–2
`unsigned`, implementation dependency 4–1
`unsigned` integer conversion 4–1
simple-escape-sequence 2–7
simple-type-specifier 7–7/8
`sin` 17–196, 201, 207
single
precision arithmetic, extension to C C–1
precision floating point arithmetic 4–1
quote 2–7
`sinh` 17–196, 202, 207
`sizeof`
array 5–11
class object 5–11
empty class 9–1
expression, implementation dependency 5–11
expression, implementation dependency type of 2–5
integral type, implementation dependency 3–11
operator 5–10/11
reference 5–11
string 2–9
type, implementation dependency 3–10
`size_t` 5–11, 17–23
implementation dependency type of 5–11
`skipws` 17–45, 58/59
`smanip<T>` 17–94/95
`smanip<T>::smanip` 17–95
source
file 2–1, 17–2, 4
file inclusion 16–3
space, white 2–2
special member function —see also constructor, destructor, inline function, user-defined conversion, virtual function
specialization 14–7
instantiation and 14–8
overloading and 14–17
template 14–9
specialized
class 14–7
function 14–7
specification, template argument 14–14
specifier
access —see access specifier
`auto` 7–2
declaration 7–2
elaborated type 3–5
`friend` 7–6
friend 17–19
function 7–4
`inline` 7–4
missing storage class 7–3
`static` 7–2
storage class 7–2
type —see type specifier

typedef 7–5
virtual 7–5
sqrt 17–196, 202, 208
<sstream> 17–4, 107
<sstream.ns> 17–4
stack unwinding 15–3
Standard
C library 17–2, 4, 12, 20, 40
C++ library 17–2, 4, 12/13, 18/20, 23, 25, 29/31, 38, 40, 48, 61, 129
standard
conversion 4–1
conversion, overloaded function and 13–6
conversion, overloading resolution and 13–5
headers 2–5
start, program 3–7/8
startup, program 17–2, 13
statement 6–1
—see also return, return
break 6–4/5
compound 6–1
continue 6–4/5
continue in for 6–4
declaration 6–5
declaration in for 6–4
declaration in switch 6–3
do 6–3/4
empty 6–1
expression 6–1
extension to C declaration C–1
for 6–3/4
goto 6–1, 4/5
if 6–2
iteration 6–3
jump 6–4
labeled 6–1
null 6–1
selection 6–2
sequence 6–1
switch 6–2/3, 5
while 6–3/4
{}, block 6–1
statement 6–1
static
NTBS 17–19
NTWCS 17–20
cast 5–7
type 1–2
static
class member 3–9
destruction of local 6–6
initialization, local 6–6
linkage of 3–6, 7–3, 21
local object 3–9
member 9–8
member declaration 3–2
member definition 9–9
member example 9–8
member function 9–6, 8
member initialization 9–9
member, linkage of 3–7
member local class restriction 9–8
member, template and 14–18
member, type of 5–10, 9–9
member use 9–8
object, destructor 3–8
object initialization 3–8, 8–12/13
objects order of execution, constructor and 12–6
objects order of execution, destructor and 12–6
overloading and 13–1
restriction 7–3
specifier 7–2
variable, template and 14–18
staticcast
class object cast 5–8
conversion to enumeration type 5–7
lvalue cast 5–8
reference cast 5–7
<stdarg.h> 8–8, 17–4, 11
__STDC__ 16–9
implementation dependency 16–9
<stddef.h> 2–5, 7, 9, 5–11, 17–4, 20, 22
stdiobuf::buffered 17–125
stdiobuf::overflow 17–125
stdiobuf::pbackfail 17–125
stdiobuf::seekoff 17–126
stdiobuf::seekpos 17–126
stdiobuf::setbuf 17–126
stdiobuf::stdiobuf 17–124
stdiobuf::~stdiobuf 17–125
stdiobuf::sync 17–126
stdiobuf::uflow 17–125
stdiobuf::underflow 17–125
stdiobuf::xsgetn 17–125
stdiobuf::xsputn 17–125
<stdio.h> 17–4, 16, 20, 22, 51, 59/61, 79, 89, 115, 117/121, 124, 128/129
<stdlib.h> 2–5, 3–7/8, 17–4, 20, 22, 33/34, 38, 40
storage
allocation new 5–11
class 3–1
class declaration 7–2
class specifier 7–2
class specifier, missing 7–3
duration 3–8
duration, auto 3–9
duration, class member 3–9
duration, local object 3–9
management —see new, delete
of array 8–8
strchr 17–21
stream
C 17–22, 51, 80, 128
buffer 17–47/48, 51, 64, 77/79, 87/89, 97, 107, 115, 124, 128/129
position 17–61/62, 65, 75/76, 103/105, 112, 119/120
<streambuf> 17–4, 59
streambuf::eback 17–68
streambuf::egptr 17–68
streambuf::epptr 17–69
streambuf::gptr 17–68
<streambuf.ns> 17–4
streambuf::overflow 17–72, 102
streambuf::pbackfail 17–73, 103
streambuf::pbase 17–68
streambuf::pbump 17–69
streambuf::pptr 17–68
streambuf::pubseekoff 17–66
streambuf::pubseekpos 17–66
streambuf::pubsetbuf 17–66
streambuf::pubsync 17–66
streambuf::sbumpc 17–66
streambuf::seekoff 17–76, 104
streambuf::seekpos 17–76, 104
streambuf::setbuf 17–77, 105, 113
streambuf::setg 17–68
streambuf::setp 17–69
streambuf::sgetc 17–67
streambuf::sgetn 17–67
streambuf::snextc 17–67
streambuf::sputbackc 17–67
streambuf::sputc 17–67

streambuf::sputn 17–67
streambuf::streambuf 17–67, 116
streambuf::~streambuf 17–66
streambuf::sungetc 17–67
streambuf::sync 17–77, 105, 113
streambuf::uflow 17–75/76, 103, 111
streambuf::underflow 17–75, 103
streambuf::xsgetn 17–76, 103, 111, 119, 125
streambuf::xsputn 17–76, 103, 112, 119, 125
streamoff 17–59, 61/62, 104, 112
streampos::offset 17–61
streampos::operator+ 17–62
streampos::operator+= 17–62
streampos::operator- 17–62
streampos::operator== 17–62
streampos::streampos 17–61
string
 concatenation 2–9
 constant 2–9
 distinct 2–9
 literal 2–9
 literal concatenation, undefined 2–9
 literal, implementation dependency 2–9
 literal, type of 2–9
 literal, undefined change to 2–9
 null-terminated byte 17–19
 null-terminated multibyte 17–19
 null-terminated wide-character 17–20
 sizeof 2–9
 terminator 0 2–9
 type of 2–9
 wide-character 2–9
string::append 17–135/136, 153, 172
string::assign 17–136/137, 154, 172
stringbuf::overflow 17–110
stringbuf::pbackfail 17–111
stringbuf::seekoff 17–112
stringbuf::seekpos 17–112
stringbuf::setbuf 17–113
stringbuf::str 17–110
stringbuf::stringbuf 17–109
stringbuf::~stringbuf 17–110
stringbuf::sync 17–113
stringbuf::uflow 17–111
stringbuf::underflow 17–111
stringbuf::xsgetn 17–111
stringbuf::xsputn 17–112
string::compare 17–145/146, 160
string::copy 17–141, 157
string::c_str 17–140
string::find 17–141/142, 157, 175
string::find_first_not_of 17–144, 159
string::find_first_of 17–142/143, 158
string::find_last_not_of 17–144/145, 159/160
string::find_last_of 17–143/144, 158/159
string::get_at 17–139, 155
<string.h> 17–4, 19/21, 100, 107, 133, 145
string::insert 17–137/138, 154, 173
string::length 17–140, 156, 175
string-literal 2–9
<string.ns> 17–4
string::operator+= 17–134/135, 153, 171
string::operator= 17–134, 153
string::operator[] 17–140, 156
string::put_at 17–139, 155
string::remove 17–138, 155, 173
string::replace 17–138/139, 155, 173
string::reserve 17–140/141, 156
string::resize 17–140, 156, 175
string::rfind 17–142, 157/158, 176
string::string 17–132/134
string::substr 17–145, 160, 176
strlen 17–99/100, 107, 133
strpbrk 17–21
strrchr 17–21
strstr 17–21
<strstream> 17–4, 97
strstreambuf::freeze 17–101
strstreambuf::overflow 17–102
strstreambuf::pbackfail 17–103
strstreambuf::pcount 17–101
strstreambuf::seekoff 17–104
strstreambuf::seekpos 17–104
strstreambuf::setbuf 17–105
strstreambuf::str 17–101
strstreambuf::strstreambuf 17–99/101
strstreambuf::~strstreambuf 17–101
strstreambuf::sync 17–105
strstreambuf::uflow 17–103
strstreambuf::underflow 17–103
strstreambuf::xsgetn 17–103
strstreambuf::xsputn 17–103
<strstream.ns> 17–4
struct 3–12
 class versus 9–1
 default member access 9–1
 initialization 8–13
 type specifier 7–9
structure 3–12, 9–1
 tag —see class name
sub-object 1–3
subscripting
 example 8–7
 explanation 8–7
 operator 5–3, 13–8
 operator, overloaded 13–10
subsequence rule, overloading 13–5
subtraction
 implementation dependency pointer 5–16
 operator 5–15
suffix
 E 2–8
 F 2–8
 L 2–7/8
 U 2–7
 f 2–8
 l 2–7/8
 u 2–7
summary
 compatibility with C C–1
 compatibility with ISO C C–2
 scope rules 10–9
 syntax A–1
switch
 statement 6–2/3, 5
 statement, declaration in 6–3
synchronization 17–48, 80, 94, 124
synonym 7–15
 type name as 7–5
syntax
 checking 14–3
 class member 5–4
 explicit instantiation 14–9
 notation 1–2
 summary A–1

T

template 14–1
 access rules 14–12
 and < 14–2

and `friend` 14–18
and `static` member 14–18
and `static` variable 14–18
argument 14–12
argument deduction 14–14
argument specification 14–14
class 14–2, 17–94/96, 162/163, 178/179, 187
definition of 14–1
function 14–13
function overloading resolution 14–14
instantiation 14–7
linkage of 3–6
member function 14–17
name injection from 14–6
name lookup 14–2
name resolution 14–2
name, use of 14–4
overload resolution 14–16
parameter, default 14–10
specialization 14–9
type equivalence 14–13
`template` 14–1
template-argument 14–2
template-argument-list 14–2
template-declaration 14–1
template-id 14–2
template-name 14–2
template-parameter 14–10
template-parameter-list 14–1
temporary 12–1
and declarator initializer 12–1
and default argument 12–1
constructor for 12–1
destruction of 12–1
destructor for 12–1
elimination of 12–1
implementation dependency generation of 12–1
order of destruction of 12–1
reference and 8–15
scope of reference 8–16
`terminate` 17–34
`terminate()` 15–5
termination
and destructor, program 12–2
program 3–7/8
terminator `0`, string 2–9
terminology, pointer 3–12
text mode 17–50
`this` 5–2
anachronism, assignment to C–11
and constructor anachronism C–11
and destructor anachronism C–11
pointer —see `this`
type of 9–7
`throw` 15–1
throw-expression in conditional-expression 5–18
throw-expression 15–1
throwing, exception 15–1
throw-point 15–1
`<time.h>` 17–4, 20
`tmpfile` 17–22
token 2–3, 5
token 2–3
translation
phases 2–1
separate 2–1
unit 17–2, 4/5, 11
unit 2–1, 3–6
unit, name and 3–1
translation@unit 2–1
trigraph 2–1/2
trivial conversions, overloading resolution 13–5
`trunc` 17–45
truncation 4–2
`try` 15–1
try-block 15–1
type 3–1
Boolean 3–10
ambiguity, declaration 7–2
arithmetic 3–11
array 3–11, 8–9
bitmask 17–18, 47, 49/50, 98
`char` 3–10
character 3–10
checking, argument 5–4
checking, extension to C C–1
checking of default argument 8–10
class and 9–1
completely-defined object 3–10
compound 3–11
`const` 7–6
conversion —see also conversion
conversion, argument 12–1
conversion, explicit —see casting
conversion rules 4–2
declaration 8–4
declaration consistency 3–7
declaration, `typedef` as 7–5
definition, class name as 9–1
`double` 3–11
dynamic 1–1
enumerated 3–10, 17–17/18, 23, 47, 51
enumeration underlying 7–11
equivalence 7–5, 9–1
equivalence, template 14–13
example of incomplete 3–10
extension to C reference C–1
extension to C user-defined C–1
`float` 3–11
floating point 3–10
function 3–11, 8–8/9
fundamental 3–10
generator —see template
implementation 17–17
implementation dependency `sizeof` 3–10
incomplete 3–10
incompletely-defined object 3–10
`int` 3–11
integral 3–10
`long` 3–11
`long double` 3–11
name 3–13, 8–2
name as synonym 7–5
name example 8–2
name example, nested 9–12
name, nested 9–12
name, scope of nested 9–12
of bit-field 9–10
of character literal 2–7
of constructor 12–1
of conversion 12–1
of `delete` 12–4
of `enum` 7–10/11
of floating point literal 2–8
of integer literal 2–7
of integer literal, implementation dependency 2–7
of `new` 12–3
of `ptrdiff_t`, implementation dependency 5–16
of `size_t`, implementation dependency 5–11
of `sizeof` expression, implementation dependency 2–5
of `static` member 5–10, 9–9

of string 2–9
of string literal 2–9
of `this` 9–7
pointer 3–11
polymorphic 10–6
`short` 3–11
`signed char` 3–10/11
specifier, `char` 7–8
specifier, `class` 7–9
specifier, `double` 7–8
specifier, elaborated 14–12
specifier, `enum` 7–9
specifier, `float` 7–8
specifier, `int` 7–8
specifier, `long` 7–8
specifier, `short` 7–8
specifier, `struct` 7–9
specifier, `union` 7–9
specifier, `unsigned` 7–8
specifier, `void` 7–8
specifier, `volatile` 7–7
static 1–2
`unsigned` 3–11
`unsigned char` 3–10/11
`unsigned int` 3–11
`unsigned long` 3–11
`unsigned short` 3–11
`void` 3–11
`void*` 3–12
`volatile` 7–6
`wchar_t` 3–11
`wchar_t` underlying 3–11
`typedef` 3–13
and `long` 7–2
and `short` 7–2
and `signed` 7–2
and `unsigned` 7–2
as type declaration 7–5
class name 7–6, 9–3
declaration 3–2
enum name 7–6
example 7–5
linkage of 3–6
overloading and 13–1
redefinition 7–5
specifier 7–5
typedef-name 7–5
`typeid` 5–7
type-id 8–2
type-id-list 15–4
`<typeinfo>` 17–4, 41
`typeinfo::name` 17–43
`<typeinfo.ns>` 17–4
`typeinfo::operator!=` 17–43
`typeinfo::operator=` 17–43
`typeinfo::operator==` 17–43
`typeinfo::typeinfo` 17–43
type-parameter 14–10
types, implementation dependency range of 2–5
type-specifier
`bool` 7–8
`wchart` 7–8
type-specifier 7–6

U

`U` suffix 2–7
`u` suffix 2–7
unary
expression 5–10
minus operator 5–10
operator 5–10
operator, interpretation of 13–9
operator, overloaded 13–9
plus operator 5–10
unary-expression 5–10
unary-operator 5–10
unbuffered file 17–128
`#undef` 16–6, 17–11
undefined
behavior 1–2
bound pointer to member function C–12
change to string literal 2–9
conversion out of range value 4–1
`delete` 5–13
deleted object 5–13
division by zero 5–1, 15
escape sequence 2–8
function call 5–7, 9
left shift 5–16
member function call 9–7
pointer comparison 5–16/17
pure virtual function call 10–9, 12–8
string literal concatenation 2–9
unrepresentable integral value 4–2
value `delete` 5–13
underlying
type, enumeration 7–11
type, `wchar_t` 3–11
underscore
character 17–11
character _ 2–4
in identifier _ 2–5
`unexpected` 17–34
`unexpected()` 15–5
`ungetc` 17–63, 67, 87, 118/119, 124
uninitialized variable, indeterminate 8–13
`union` 3–12, 9–9
access control, anonymous 9–10
anonymous 9–9
`class` versus 9–1
constructor 9–9
default member access 9–1
destructor 9–9
extension to C anonymous C–1
global anonymous 9–10
initialization 8–14, 9–9
member function 9–9
restriction 9–9, 12–1
restriction, anonymous 9–10
type specifier 7–9
unit, translation 17–2, 4/5, 11
`unitbuf` 17–45
unknown argument type 8–8
unnamed
bit-field 9–10
class 7–6
object 12–1
parameter example 8–12
unqualified-id 5–2
unrepresentable integral value, undefined 4–2
unreserved name 17–16
`unsigned`
arithmetic 3–11
`char` type 3–10/11
constant 2–6
implementation dependency `signed` 4–1
`int` type 3–11
integer conversion, `signed` 4–1
literal 2–6/7
`long` type 3–11

`short` type 3–11
type 3–11
type specifier 7–8
`typedef` and 7–2
unsigned-suffix 2–6
unspecified
argument to constructor 5–13
behavior 1–2
destructor call 6–6
expression 5–4
order of argument evaluation 5–4
order of evaluation 5–1
order of evaluation `new` 5–13
order of function call evaluation 5–4
value `new` 5–12
unwinding, stack 15–3
uppercase 17–11, 19, 45, 49, 58/59, 89/90
`uppercase` 17–45, 58/59
use of template name 14–4
user-defined
conversion 5–1, 12–1
conversion and name hiding 12–1
conversion, implicit 12–1
conversion, inheritance of 12–1
conversion, overloading resolution and 13–6
conversion, virtual 12–1
type, extension to C C–1
using-declaration 7–15
using-directive 7–18

V

`va_end` 17–11
`va_list` 17–11
value
call by 5–4
`delete`, undefined 5–13
`new`, unspecified 5–12
of `char` literal, implementation dependency 2–8
of enumerator 7–10
of multicharacter literal, implementation dependency 2–7
undefined conversion out of range 4–1
undefined unrepresentable integral 4–2
variable
argument list 8–8
indeterminate uninitialized 8–13
parameter list 5–4, 8–8
parameter list example 8–10
template and `static` 14–18
vertical tab 2–7
virtual
base class 10–3
base class DAG 10–3
base class dominance 10–5
base class initialization 12–6, 8
base class initialization, order of 12–6
destructor 12–2
destructor, pure 12–2
function 10–6
function access 11–7
function call 10–8
function call, constructor and 12–7
function call, destructor and 12–7
function call, undefined pure 10–9, 12–8
function definition 10–7
function definition, pure 10–8
function example 10–6/7
function example, pure 10–9
function, pure 10–8/9
member function 17–13, 15, 76
user-defined conversion 12–1
`virtual`
and `friend` 10–7
and multiple inheritance 10–7
specifier 7–5
visibility 3–5
`void`
parameter 8–8
type 3–11
type specifier 7–8
`void&` 8–5
`void*`
conversion, pointer to member 4–4
pointer comparison 5–16
pointer conversion 4–2
pointer type extension to C C–1
type 3–12
`volatile` 3–12
assignment, pointer to 5–19
`const void*` pointer conversion 4–2
constructor and 9–8, 12–1
destructor and 9–8, 12–1
extension to C C–2
implementation dependency 7–7
initialization, pointer to 8–12
member function 9–7/8
operand 5–1
overloading and 13–1/2
pointer assignment 5–19
pointer initialization 8–12
reference 8–16
type 7–6
type specifier 7–7
`void*` pointer conversion 4–2

W

`<wchar.h>` 17–4, 12, 20, 23, 152, 160
`wchart` type-specifier 7–8
`wchar_t` 2–7, 9, 17–19, 23
implementation dependency 3–11
implementation dependency promotion of 4–1
type 3–11
underlying type 3–11
`wcslen` 17–152
`<wctype.h>` 17–4, 12
`WEOF` 17–23
`while` statement 6–3/4
white
space 2–2
space 2–3
wide-character 2–7
string 2–9
string, null-terminated 17–20
`wint_t` 17–23
`wmemcmp` 17–160
`ws` 17–58/59, 81, 87
`<wstring>` 17–4, 148
`wstring::append` 17–153
`wstring::assign` 17–154
`wstring::compare` 17–160
`wstring::copy` 17–157
`wstring::c_wcs` 17–156
`wstring::find` 17–157
`wstring::find_first_not_of` 17–159
`wstring::find_first_of` 17–158
`wstring::find_last_not_of` 17–159/160
`wstring::find_last_of` 17–158/159
`wstring::get_at` 17–155

wstring::insert 17–154
wstring::length 17–156
<wstring.ns> 17–4
wstring::operator+= 17–153
wstring::operator= 17–153
wstring::operator[] 17–156
wstring::put_at 17–155
wstring::remove 17–155
wstring::replace 17–155
wstring::reserve 17–156
wstring::resize 17–156
wstring::rfind 17–157/158
wstring::substr 17–160
wstring::wstring 17–151/152

X

xalloc::do_raise 17–36
xalloc::xalloc 17–36
xalloc::~xalloc 17–36
xdomain::do_raise 17–32
xdomain::xdomain 17–32
xdomain::~xdomain 17–32
xlogic::do_raise 17–27
xlogic::xlogic 17–27
xlogic::~xlogic 17–27
xmsg::do_raise 17–26/33, 36, 42, 49
xmsg::raise 17–26
xmsg::raise_handler 17–25
xmsg::set_raise_handler 17–26
xmsg::what 17–26
xmsg::where 17–26
xmsg::why 17–26
xmsg::xmsg 17–26/27
xmsg::~xmsg 17–26
xrange::do_raise 17–33
xrange::xrange 17–33
xrange::~xrange 17–33
xruntime::do_raise 17–28
xruntime::xruntime 17–28
xruntime::~xruntime 17–28
X(X&) —see copy constructor

Z

zero
assignment to pointer 5–19
assignment to pointer to member 5–19
pointer 4–2/3, 5–16
pointer conversion 4–3
undefined division by 5–1, 15
width of bit-field 9–10