Index

, —see comma operator
! —see logical negation operator
!= —see inequality operator
operator 16−5
operator 16−6
% —see modulus operator
%= operator 5−22
&
 —see address-of operator
 —see bitwise AND operator
 reference declarator 8−5
&& —see logical AND operator
&= operator 5−22
()
 —see function call operator
 function declarator 8−8
*
 —see indirection operator
 —see multiplication operator
 pointer declarator 8−4
*= operator 5−22
+
 —see addition operator
 —see unary plus operator
++ —see increment operator
+= operator 5−13, 22
−
 —see subtraction operator
 —see unary minus operator
−− —see decrement operator
−= operator 5−22
−> —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 5−22
:
 field declaration 9−11
 label specifier 6−1
::
 —see scope resolution operator
 scope resolution operator 3−6
::*, pointer to member declarator 8−6
<
 —see less than operator
 template and 14−2
<< —see left shift operator
<<= operator 5−22
<= —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 5−22
?: —see conditional expression operator
[]
 —see subscripting operator
 array declarator 8−7
\ —see backslash
^ —see bitwise exclusive OR operator
^= operator 5−22
_
 underscore character 2−4
 underscore in identifier 2−5
{}
 block statement 6−1
 class declaration 9−1
 class definition 9−1
 enum declaration 7−11
 initializer list 8−15
| —see bitwise inclusive OR operator
|= operator 5−22
|| —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 3−10, 6−5, 18−14, 17
abs 26−21
abstract
 class 10−9
 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 11−4
base class 11−2
base class member 10−1
class member 5−6
control 11−1
control, anonymous `union` 9−11
control default 11−1
control, member function and 12−1
control, overloading resolution and 10−4
declaration 11−3
example, member name 11−3
member name 11−1
overloading and 13−4
`protected` member 11−6
rules, template 14−18
specifier 11−2
specifier and `friend` 11−5
specifier and object layout 11−2
`struct` default member 9−1
`union` default member 9−1
virtual function 11−7
access-specifier 10−1
`accumulate` 26−31
`acos` 26−21
addition operator 5−18
additive operator 5−18
additive-expression 5−18
address
of bit-field 9−11
of bit-field restriction 9−11
of constructor 12−2
of overloaded function 5−13, 13−15
of qualified name 5−12
address-of operator 5−12
`adjacent_find` 25−10
adjusting base class member access 11−3
adjustment
array parameter 8−9
function parameter 8−9
`advance` 24−13
aggregate 8−15
alert 2−7
`<algorithm>` 25−1
alias 7−17
alignment
of bit-field 9−11
of bit-field, implementation dependency 9−11
requirement, implementation dependency 3−13
`allocate` 20−16
allocation
function 3−12, 5−15, 12−8
implementation dependency 9−4, 11−2
implementation dependency bit-field 9−11
`new`, storage 5−14
Allocator Requirements 20−1
`allocator` 20−12
allowing an exception 15−6
altermate definition 17−19
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 13−4
function declaration 8−14
`if-else` 6−2
member access 10−3
parentheses and 5−14
resolution, scoping 10−4
Amendment 1 C−13
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 5−20
operator, logical 5−21
operator, side effects and logical 5−21
anonymous
`union` 9−10
`union` access control 9−11
`union`, extension to C C−1
`union`, global 9−11
`union` restriction 9−11
`any`, `bits` 23−13
`append`, `basic_string` 21−10
`apply`, `valarray` 26−22
`argc` 3−9
argument 1−2, 17−19/20, 19−4/C−13
and name hiding, default 8−12
and virtual function, default 8−12
binding of default 8−11
conversion 5−5, 8−9
declaration, default 8−10
deduction, template 14−20
evaluation of default 8−11/12
evaluation, order of 5−5
evaluation, unspecified order of 5−5
example of default 8−10/11
list, empty 8−8
list, variable 8−8
matching —see overload resolution
overloaded operator and default 13−16
overloading and default 8−12
passing 5−5
passing, reference and 8−17
reference 5−5
scope of default 8−12
specification, template 14−19
substitution 16−5
template 14−17
to constructor, unspecified 5−16
type checking 5−5
type checking of default 8−11
type conversion 12−3
type, unknown 8−8
`argv[]` 3−9
arithmetic
conversions, usual 5−2
exception 5−1
exception, implementation dependency 5−1
extension to C single precision C−1
pointer 5−18
type 3−15
`unsigned` 3−15
array
bound 8−7

const 3−16
constructor and 5−15
declaration 8−7
declarator [] 8−7
declarator, multidimensional 8−7
default constructor and 5−15
delete 5−16
example 8−7
initialization 8−15
member 9−4
multidimensional 8−8
new 5−15
of class objects and constructor 12−10
of class objects and new 5−15
of class objects initialization 8−16, 12−10
order of execution, constructor and 12−9
order of execution, destructor and 12−6
overloading and pointer versus 13−2
parameter adjustment 8−9
pointer conversion 4−1
size, default 8−7
sizeof 5−13
storage of 8−8
type 3−15, 8−9
array-to-pointer conversion 4−1
arrow operator —see class member access operator
asin 26−21
asm
declaration 7−23
implementation dependency 7−23
assembler 7−23
<assert.h> 17−18/C−12
assign
basic_string 21−10
deque 23−15
list 23−18
string 21−21
string_char_traits 21−4
vector 23−24
wstring 21−23
assignment
and initialization, overloaded 12−10
and lvalue 5−22
base class object 5−22
conversion by 5−22
derived class object 5−22
expression 5−22
extension to C memberwise C−2
operator 5−22, 17−7
operator, copy 12−17
operator, overloaded 13−17
operator restriction, copy 12−18
reference 8−17
to class object 5−22
to reference 5−22
to this anachronism C−11
virtual base class 12−18
assignment-expression 5−22
assignment-operator 5−22
at, basic_string 21−12
atan 26−21
atan2 26−21
atexit 3−10, 6−6, 18−10
auto
destruction of 6−4/5
initialization 6−6
object initialization 8−14
restriction 7−3
specifier 7−2
storage duration 3−12
automatic initialization 6−5/6
auto_ptr 20−17
auto_ptr 20−17
auto_ptr 20−17
operator* 20−18
operator-> 20−18
release 20−18
reset 20−18
~auto_ptr 20−18
~auto_ptr, auto_ptr 20−18

B

back_inserter 24−19
back_insert_iterator 24−18
back_insert_iterator 24−19
back_insert_iterator 24−19
operator* 24−19
operator++ 24−19
operator= 24−19
backslash character 2−7
backspace 2−7
bad, ios_base 27−17
bad_alloc 5−16, 18−14
bad_alloc 18−14
bad_alloc 18−14
what 18−14
bad_cast 5−8, 18−16
bad_cast 18−17
bad_cast 18−17
bad_typeid 5−8, 18−17
bad_typeid 18−17
bad_typeid 18−17
base
class 17−7, 19/20, 18−14
class 10−1/2
class access 11−2
class cast 5−10
class constructor order of execution 12−2
class conversion 4−4
class destructor order of execution 12−6
class, direct 10−1
class, indirect 10−1
class initialization 12−10
class initialization, order of 12−11
class initializer 8−13
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 5−22
class, private 11−2
class, public 11−2
class virtual —see virtual base class
of integer literal 2−6
base-specifier 10−1
base-specifier-list 10−1
basic_convbuf 27−67
basic_convbuf 27−67
basic_convbuf 27−67
overflow 27−68
pbackfail 27−68
seekoff 27−69
seekpos 27−70
setbuf 27−70
sync 27−70
underflow 27−68
~basic_convbuf 27−68
~basic_convbuf, basic_convbuf 27−68
basic_filebuf 27−74
basic_filebuf 27−75

basic_filebuf 27–75
close 27–76
is_open 27–75
open 27–75
overflow 27–76
pbackfail 27–77
seekoff 27–78
seekpos 27–78
setbuf 27–79
showmany 27–77
sync 27–79
underflow 27–78
~basic_filebuf 27–75
~basic_filebuf, basic_filebuf 27–75
basic_filebuf<char> 27–73
basic_filebuf<wchar_t> 27–73
basic_ifstream 27–79
basic_ifstream 27–80
basic_ifstream 27–80
close 27–80
is_open 27–80
open 27–80
rdbuf 27–80
basic_ifstream<char> 27–73
basic_ifstream<wchar_t> 27–73
basic_ios 27–20
basic_ios 27–21
basic_ios 27–21
fill 27–18
init 27–36, 46
rdbuf 27–22
tie 27–21
basic_ios<char> 27–4
basic_ios<char> 27–7
basic_ios<wchar_t> 27–4
basic_ios<wchar_t> 27–7
basic_istream 27–34
basic_istream 27–35
basic_istream 27–35
gcount 27–44
get 27–41
getline 27–42
ignore 27–43
ipfx 27–36
isfx 27–37
operator>> 27–39
peek 27–44
putback 27–44
read 27–43
readsome 27–44
sync 27–37
unget 27–44
basic_istream<char> 27–4
basic_istream<char> 27–33
basic_istream<wchar_t> 27–4
basic_istream<wchar_t> 27–33
basic_istringstream 27–59
basic_istringstream 27–60
basic_istringstream 27–60
rdbuf 27–60
str 27–60
basic_istringstream<char> 27–54
basic_istringstream<wchar_t> 27–54
basic_ofstream 27–80
basic_ofstream 27–81
basic_ofstream 27–81
close 27–82
is_open 27–81
open 27–82
rdbuf 27–81
basic_ofstream<char> 27–73
basic_ofstream<wchar_t> 27–73
basic_ostream
basic_ostream 27–46
basic_ostream 27–46
flush 27–46
operator<< 27–49
opfx 27–46
osfx 27–46
put 27–51
write 27–51
~basic_ostream 27–46
~basic_ostream, basic_ostream 27–46
basic_ostream<char> 27–4
basic_ostream<char> 27–33
basic_ostream<wchar_t> 27–4
basic_ostream<wchar_t> 27–33
basic_ostringstream 27–60
basic_ostringstream 27–61
basic_ostringstream 27–61
rdbuf 27–61
str 27–62
basic_ostringstream<char> 27–54
basic_ostringstream<wchar_t> 27–54
basic_streambuf 27–24
basic_streambuf 27–27
basic_streambuf 27–27
eback 27–29
egptr 27–29
epptr 27–29
gbump 27–29
gptr 27–29
in_avail 27–27
overflow 27–30
pbackfail 27–30
pbase 27–29
pbump 27–29
pptr 27–29
pubseekoff 27–27
pubseekpos 27–27
pubsetbuf 27–27
pubsync 27–27
sbumpc 27–28
seekoff 27–32
seekpos 27–32
setbuf 27–33, 59
setg 27–29
setp 27–30
sgetc 27–28
sgetn 27–28
showmany 27–31, 77
snextc 27–28
sputbackc 27–28
sputc 27–28
sputn 27–28
sungetc 27–28
sync 27–33
uflow 27–32
underflow 27–31
xsgetn 27–32
xsputn 27–32
basic_streambuf<char> 27–24
basic_streambuf<wchar_t> 27–24
basic_string 21–5, 7, 18
append 21–10
assign 21–10
at 21–12
basic_string 21–7
basic_string 21–7
c_str 21–13
capacity 21–14
compare 21–18

copy 21−14
data 21−13
empty 21−14
find 21−15
find_first_not_of 21−16
find_first_of 21−15
find_last_not_of 21−17
find_last_of 21−16
insert 21−11
max_size 21−13
operator!= 21−19
operator+ 21−18
operator+= 21−9
operator< 21−20
operator<< 21−21
operator<= 21−20
operator= 21−9
operator== 21−19
operator> 21−20
operator>= 21−21
operator>> 21−21
operator[] 21−12
remove 21−11
replace 21−12
resize 21−13
rfind 21−15
size 21−13
substr 21−17
swap 21−14
basic_stringbuf 27−54
basic_stringbuf 27−56
basic_stringbuf 27−56
overflow 27−57
pbackfail 27−57
seekoff 27−57
seekpos 27−58
setbuf 27−59
str 27−56
underflow 27−57
basic_stringbuf<char> 27−54
basic_stringbuf<wchar_t> 27−54
before, type_info 18−16
behavior
default 17−1, 4
implementation-defined 1−2
locale-specific 1−2
reentrancy, implementation-defined 17−20
required 17−2, 4
undefined 1−2
unspecified 1−2
Ben 13−3
bidirectional_iterator 24−11
distance_type 24−13
iterator_category 24−12
value_type 24−12
bidirectional_iterator_tag 24−11
binary
operator, interpretation of 13−17
operator, overloaded 13−17
binary_function 20−6
binary_negate 20−9
binary_search 25−22
bind1st 20−10
bind2nd 20−10
binder1st 20−9
binder2nd 20−10
binding
—see virtual function, dynamic
of default argument 8−11
reference 8−17
bit-field 9−11
address of 9−11
alignment of 9−11
allocation, implementation dependency 9−11
declaration 9−11
implementation dependency alignment of 9−11
implementation dependency sign of 9−11
layout 9−11
restriction 9−11
restriction, address of 9−11
restriction, pointer to 9−11
type of 9−11
unnamed 9−11
zero width of 9−11
bit-fields, Boolean 3−15
bitmask type 17−5
bitmask
operator& 17−5
operator&= 17−5
operator^ 17−5
operator^= 17−5
operator| 17−5
operator|= 17−5
operator~ 17−5
<bits> 23−7
bits 23−8
any 23−13
bits 23−9
bits 23−9
count 23−12
length 23−12
none 23−13
operator!= 23−12
operator& 23−13
operator&= 23−10
operator<< 23−13/14
operator<<= 23−10
operator== 23−12
operator>> 23−13
operator>>= 23−10
operator^ 23−13
operator^= 23−10
operator| 23−13
operator|= 23−10
operator~ 23−11
reset 23−11
set 23−11
test 23−12
to_string 23−12
to_ulong 23−12
to_ushort 23−11
toggle 23−11
bitwise
AND operator 5−20
exclusive OR operator 5−20
inclusive OR operator 5−20
operator 5−20
block
initialization in 6−5
scope —see local scope
statement {} 6−1
structure 6−5
body, function 8−13
bool promotion to int 4−2
bool
increment 5−7, 13
type-specifier 7−8
boolalpha 27−22
Boolean
bit-fields 3−15
constant 2−9
conversion 4−4

literal 2−9
type 3−14
type 3−15
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
built-in type —see fundamental type
byte 5−13
string, null-terminated 17−6

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 17−7, 19, 18−2/C−14
headers, ISO 2−5
implementation dependency extension to C−10
inline function, extension to C−1
library, Standard 17−1, 6, 17−17/C−12, C−14
linkage to 7−23
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
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 5−5
by value 5−5
operator function 13−16
`calloc` 20−18
`capacity`
`basic_string` 21−14
`vector` 23−24
carriage return 2−7
`case` label 6−1, 3
`<cassert>` 17−18, 19−6
cast
ambiguity, declaration versus 8−3
base class 5−10
const 5−11
derived class 5−10
dynamic 5−7, 18−16
implementation dependency pointer to function 5−10
integer to pointer 5−10
lvalue 5−9/10
of pointer to member anachronism C−12
operator 5−12, 17, 8−2
pointer to function 5−10
pointer to integer 5−10
pointer to member 5−10
reference 5−9, 11
reinterpret 5−10
`reinterpret_cast`, lvalue 5−10
`reinterpret_cast`, reference 5−11
static 5−9
`static_cast`, lvalue 5−9
`static_cast`, reference 5−9
to incomplete class 5−10
cast-expression 5−17
casting 5−5, 17
`catch` 15−1
category 17−1
`category`, `locale` 22−4
c-char 2−7
c-char-sequence 2−6
`<cctype>` 21−24
`cerr` 27−5
`<cerrno>` 17−19, 19−6
`<cfloat>` 17−18, 18−10
C++
Standard library 17−1, 19/21
header 17−17/C−12
library, Standard C−12
change
to `const` object, undefined 7−7
to string literal, undefined 2−8
`char`
implementation dependency sign of 3−14
literal, implementation dependency value of 2−8
type 3−14
type, `signed` 3−14
type specifier 7−8
type, `unsigned` 3−14
character
array initialization 8−16
constant 2−7
decimal-point 17−6
literal 2−7
literal, type of 2−7
multibyte 1−2
`signed` 3−14
string 2−8
type 3−14
underscore 17−19
character-literal 2−6
`char_in`, `string_char_traits` 21−4
`char_out`, `string_char_traits` 21−4
`CHAR_T` 27−2
checking
point of error 14−4
syntax 14−4
`cin` 27−5
class 3−15, 9−1
abstract 10−9
access and `friend` 11−5
anachronism, nonnested C−12
and type 9−1
base 17−7, 19/20, 18−14
base —see base class
cast to incomplete 5−10
constructor and abstract 10−9
conversion 12−3
conversion ambiguity 10−5
conversion, base 4−4
declaration, forward 9−2, 10−1
declaration `{}` 9−1

definition 9–1, 4
definition 3–3
definition example 9–4
definition name hiding 9–2
definition, scope of 9–2
definition {} 9–1
derived 17–20
derived —see derived class
extension to C C–1
friend 11–5
generated 14–9
gslice 26–24
linkage of 3–8
linkage specification 7–23
local —see local class
member —see also member
member access 5–6
member access operator 5–6
member declaration 9–3
member function 9–6
member initialization 8–14
member semantics 5–6
member, static 3–11
member storage duration 3–13
member syntax 5–6
name 8–2
name as type definition 9–2
name declaration 3–2
name, elaborated 7–9, 9–2/3
name, point of declaration 9–3
name, scope of 9–2
name, typedef 7–6, 9–3
nested —see nested class
object, assignment to 5–22
object, const 3–16
object copy 12–16
object copy —see also copy constructor
object initialization 8–15, 12–9
object initialization —see also constructor
object layout 9–4, 10–2
object, member 9–4
object, operations on 9–1
object, sizeof 5–13
objects and constructor, array of 12–10
objects and new, array of 5–15
objects initialization, array of 8–16, 12–10
pointer to abstract 10–9
polymorphic 10–6
scope 3–5
scope of enumerator 7–13
scope of friend 11–5
sizeof, empty 9–1
specialized 14–9
template 14–2, 23–9
type restriction, member of 12–11
unnamed 7–6
class
type specifier 7–9
versus struct 9–1
versus union 9–1
classic, locale 22–7
class-key 7–9, 9–1
class-name 9–1
class-specifier 9–1
clear, ios_base 27–17
<climits> 3–14, 17–18, 18–9, 23–9
<clocale> 17–6, 22–29/C–13
clog 27–6
close
basic_filebuf 27–76
basic_ifstream 27–80
basic_ofstream 27–82
<cmath> 26–33
codecvt 22–20
convert 22–20
codecvt_byname 22–21
collate 22–18
compare 22–19
hash 22–19
transform 22–19
collate_byname 22–19
operator() 22–19
comma
operator 5–22
operator, side effects and 5–22
comment 2–3
/* */ 2–4
// 2–4
compare
basic_string 21–18
collate 22–19
string 21–22
string_char_traits 21–5, 18
wstring 21–23
comparison
function 17–1
implementation dependency pointer 5–20
pointer 5–19/20
pointer to function 5–19
undefined pointer 5–19/20
void* pointer 5–19
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 3–13
<complex> 26–1, 3
complex 26–3
complex 26–6
complex 26–6
exp 26–9
imag 26–9
log 26–9
norm 26–9
operator!= 26–8
operator* 26–7
operator*= 26–7
operator+ 26–7
operator+= 26–6
operator- 26–7
operator-= 26–6
operator/= 26–7
operator<< 26–9
operator== 26–8
operator>> 26–8
polar 26–9
pow 26–9
real 26–10
sin 26–10
sinh 26–10
sqrt 26–10
component 17–1
compound
statement 6–1
type 3–15
compound-statement 6–1
concatenation
string 2–9
undefined string literal 2–9
condition 6–2

conditional
  expression operator 5−21
  inclusion 16−2
conditional-expression, throw-expression in 5−21
conditions, rules for 6−2
consistency
  example, linkage 7−3
  linkage 7−3
  linkage specification 7−23
  type declaration 3−9
const cast 5−11
*const example 8−4
const 3−16
  array 3−16
  class object 3−16
  constructor and 9−8, 12−1
  destructor and 9−8, 12−6
  example 8−4
  extension to C C−1
  initialization 7−7, 8−15
  linkage of 3−8, 7−3
  member function 9−8
  member initialization 12−10
  object, undefined change to 7−7
  operand 5−1
  overloading and 13−1/2
  reference 8−18
  type 7−7
constant 2−5, 3−15, 5−3
  character 2−7
  enumeration 7−11
  expression 5−23
  expression, pointer to member 5−12
  initializer 9−3
  integer 2−6
  long 2−6
  null pointer 4−3/4
  pointer declaration 8−4
  pointer example 8−4
  unsigned 2−6
constant-expression 5−23
constant-initializer 9−3
construct 20−16
constructor 12−1
  address of 12−2
  anachronism, free store and C−11
  anachronism, this and C−11
  and abstract class 10−9
  and array 5−15
  and array order of execution 12−9
  and const 9−8, 12−1
  and initialization 12−9
  and initialization example 12−9
  and member function 12−2
  and new 5−15
  and new, implementation dependency 5−16
  and return 6−5
  and static objects order of execution 12−10
  and virtual function call 12−14
  and volatile 9−8, 12−1
  array of class objects and 12−10
  call, explicit 12−2
  conversion by 12−3
  conversion by —see also user-defined conversion
  copy 12−2, 16, 17−7
  default 26−22
  default —see default constructor
  definition 8−13
  example 12−2
  exception handling 15−3
  for temporary 12−2
  inheritance of 12−1
  non-trivial 12−1
  order of execution, base class 12−2
  order of execution, member 12−2
  restriction 12−1/2
  restriction, copy 12−17
  type of 12−2
  union 9−10
  unspecified argument to 5−16
Container Requirements 23−1
continue
  in for statement 6−4
  statement 6−4/5
control line —see preprocessing directive
convention 17−4
conversion
  Boolean 4−4
  ambiguity, class 10−5
  anachronism, pointer to member C−12
  and name hiding, user-defined 12−5
  argument 5−5, 8−9
  array pointer 4−1
  array-to-pointer 4−1
  base class 4−4
  by assignment 5−22
  by constructor 12−3
  class 12−3
  explicit type —see casting
  floating point 4−3
  floating-integral 4−3
  function —see also user-defined conversion
  function-to-pointer 4−2
  implementation defined pointer integer 5−10
  implementation dependency floating point 4−3
  implicit 4−1, 5−1, 12−3
  implicit user-defined 12−5
  inheritance of user-defined 12−5
  integer 4−3
  lvalue-to-rvalue 4−1
  operator 5−1, 12−4
  overload resolution and 13−10
  overload resolution and pointer 13−15
  pointer 4−3
  pointer to function 4−2
  pointer to member 4−4
  pointer to member void* 4−4
  rank 13−12
  return type 6−5
  reverse_bidirectional_iterator 24−14
  reverse_iterator 24−17
  sequence, implicit 4−1
  signed unsigned integer 4−3
  standard 4−1
  to enumeration type 5−9
  to enumeration type, static_cast, 5−9
  to rvalue, lvalue 4−1
  type of 12−4
  undefined floating point 4−3
  user-defined 4−1, 5−1, 12−3/4
  virtual user-defined 12−5
conversion-function-id 12−4
conversions
  qualification 4−2
  usual arithmetic 5−2
convert, codecvt 22−20
convin, ios_conv_traits<STATE_T> 27−64
convout, ios_conv_traits<STATE_T> 27−65
conv_traits 27−63
copy
  assignment operator 12−17
  assignment operator 12−16

assignment operator, implicitly-declared 12–17
assignment operator restriction 12–18
class object 12–16
constructor 12–2, 16, 17–7
constructor, implicitly-declared 12–16
constructor restriction 12–17
copy 25–13
basic_string 21–14
ios_traits 27–10
string 21–22
string_char_traits 21–5
wstring 21–23
copy_backward 25–13
copyfmt, ios_base 27–16
cos 26–21
cosh 26–21
count 25–11
bits 23–12
count_if 25–11
cout 27–5
__cplusplus 16–9
<csetjmp> 17–19, 18–19
<csignal> 18–19
<cstdarg> 8–8, 17–18/19, 18–19
<cstddef> 5–14, 19, 17–18, 18–2/C–13
<cstdio> 17–8, 27–5/6, 38, 47, 73, 76, 78, 27–82/C–13
<cstdlib> 3–10, 17–18, 18–10, 19, 20–12, 21–25, 25–29, 26–33, 27–54/C–13
c_str, basic_string 21–13
<cstream> 27–62
<cstring> 17–6, 20–12, 21–24/C–13
<ctime> 18–19, 20–18, 22–2/C–13
ctor-initializer 12–10
ctype 22–9
ctype 22–12
ctype 22–12
do_is 22–11
do_narrow 22–11
do_scan_is 22–11
do_scan_not 22–11
do_tolower 22–11
do_toupper 22–11
do_widen 22–11
is 22–11
narrow 22–12
scan_is 22–12
scan_not 22–12
tolower 22–12
toupper 22–12
widen 22–12
ctype<char>
ctype<char> 22–15
ctype<char> 22–15
is 22–14
narrow 22–15
scan_is 22–14
scan_not 22–14
tolower 22–15
toupper 22–14
widen 22–15
~ctype<char> 22–15
~ctype<char>, ctype<char> 22–15
<ctype.h> C–12
ctype_name 22–12
cv-qualifier 3–16
cv-qualifier 8–2
<cwchar> 17–7, 18–3, 21–24, 27–82/C–13
<cwctype> 21–24/C–13

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
member, static 9–9
data, basic_string 21–13
deallocate 20–16
deallocation
—see delete
function 3–13, 5–17, 12–8
dec 27–23, 39, 49
decimal literal 2–6
decimal-literal 2–5
decimal-point character 17–6
declaration 3–1/2, 7–1
:, field 9–11
access 11–3
ambiguity, function 8–14
array 8–7
as definition 7–2
asm 7–23
bit-field 9–11
class member 9–3
class name 3–2
class name, point of 9–3
consistency, type 3–9
constant pointer 8–4
default argument 8–10
definition versus 3–2
ellipsis in function 5–5, 8–8
enumerator, point of 3–6
example 3–2, 8–9
example, function 8–10
extern 3–2
extern reference 8–17
forward 7–4
forward class 9–2, 10–1
function 3–2, 8–8
function template 14–24
hiding —see name hiding
in for, scope of 6–4
in for statement 6–4
in switch statement 6–3
matching, overloaded function 13–3
member 9–3
multiple 3–9
name 3–2
name, point of 3–6
overloaded 13–1
overloaded name and friend 11–4
parameter 8–8/9
parentheses in 8–3/4
pointer 8–4
reference 8–5
register 7–3
scope of friend 3–5
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 7–11
declaration 7–1
declaration-seq 7–23
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 12–3
meaning of 8–4
multidimensional array 8–7
declarator 8–1
declarator-id 8–2
decl-specifier 7–2
decrement
operator 5–7, 12/13
operator, overloaded 13–17
deduction, template argument 14–20
default
access control 11–1
argument and name hiding 8–12
argument and virtual function 8–12
argument, binding of 8–11
argument declaration 8–10
argument, evaluation of 8–11/12
argument, example of 8–10/11
argument, overload resolution and 13–9
argument, overloaded operator and 13–16
argument, overloading and 8–12
argument, scope of 8–12
argument, type checking of 8–11
array size 8–7
behavior 17–1, 4
constructor 26–22
constructor 12–1
constructor and array 5–15
constructor and initialization 12–9
constructor and new 5–15
destructor 12–6
initialization 8–14
initializers, overloading and 13–3
member access, struct 9–1
member access, union 9–1
default label 6–1, 3
#define 16–5
definition 3–2, 17–1
altermate 17–19
and initialization 7–2
class 3–3
class 9–1, 4
class name as type 9–2
constructor 8–13
declaration as 7–2
enumerator 3–3
enumerator point of 7–12
example 3–2
example, function 8–13
example, nested class 9–12
function 3–3
function 8–13
function template 14–24
local class 9–13
member function 9–6
name hiding, class 9–2
namespace 7–13
nested class 9–11
object 3–3
of template 14–1
pure virtual function 10–9
scope, macro 16–6
scope of class 9–2
static member 9–10
versus declaration 3–2
virtual function 10–7
{}, class 9–1
definitions, implementation-generated 3–2
delete
array 5–16
object 5–16
delete 3–12, 5–16/17, 12–8
destructor and 5–17, 12–7
example 12–8
example, destructor and 12–9
example, scope of 12–9
extension to C C–1
extension to C overloading C–2
operator 17–19, 18–12, 20–18
overloading and 3–13
type of 12–8
undefined 5–16
undefined value 5–16
delete[], operator 17–19, 18–13
deleted object, undefined 3–13, 5–16
delete-expression 5–16
dependent name 14–6
deprecated features 5–7, 13
<deque> 23–7
deque 23–14
assign 23–15
erase 23–16
insert 23–15
resize 23–16
dereferencing 5–1
—see also indirection
derivation —see inheritance
derived
class 17–20
class 10–1
class cast 5–10
class example 10–1
class, most 12–11
class object, assignment 5–22
class, overloading and 13–3
destination type 8–15
destroy 20–16
destruction
of auto 6–4/5
of local static 6–6
of local variable 6–4/5
of temporary 12–2
of temporary, order of 12–3
destructor 12–6, 17–7
anachronism, free store and C–11
anachronism, this and C–11
and array order of execution 12–6
and const 9–8, 12–6
and delete 5–17, 12–7
and delete example 12–9
and exception, explicit 12–7
and exit from scope 6–4
and fundamental type 12–7
and member function 12–6
and placement of object 12–7
and virtual function call 12–14
and volatile 9–8, 12–6
call example, explicit 12–7

call, explicit 12−7
call, implicit 12−7
call, unspecified 6−6
default 12−6
exception handling 15−3
extension to C C−2
for temporary 12−2
inheritance of 12−6
non-trivial 12−6
order of execution 12−6
order of execution, base class 12−6
order of execution, member 12−6
program termination and 12−7
pure virtual 12−6
restriction 12−6
`static` object 3−10
`union` 9−10
virtual 12−6
diagnostic message 1−2
difference from C expression evaluation C−1
digit 2−4
digit-sequence 2−8
digraph 2−3
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 5−14
`distance` 24−13
`distance_type`
`T*` 24−13
`bidirectional_iterator` 24−13
`forward_iterator` 24−13
`input_iterator` 24−13
`random_access_iterator` 24−13
distinct string 2−8
`divides` 20−7
division
by zero, undefined 5−1, 18
implementation dependency 5−18
operator 5−18
`djacent_difference` 26−32
`do` statement 6−3/4
`do_is`, `ctype` 22−11
`domain_error` 19−3
`domain_error` 19−3
`domain_error` 19−3
dominance, virtual base class 10−5
`do_narrow`, `ctype` 22−11
`do_scan_is`, `ctype` 22−11
`do_scan_not`, `ctype` 22−11
dot operator —see class member access operator
`do_tolower`, `ctype` 22−11
`do_toupper`, `ctype` 22−11
double quote 2−7
`double`
literal 2−8
type 3−15
type specifier 7−8
`do_widen`, `ctype` 22−11
dynamic
binding —see virtual function
cast 5−7, 18−16
initialization 3−10
storage duration 3−12, 5−14
type 1−2

E

`E` suffix 2−8
`eback`, `basic_streambuf` 27−29
`egptr`, `basic_streambuf` 27−29
elaborated
class name 7−9, 9−2/3
`enum` name 7−9
type specifier 14−5
type specifier 3−6
type specifier —see elaborated class name
elaborated-type-specifier 7−9
`#elif` 16−2
elimination of temporary 12−2
ellipsis
example 8−10
in function declaration 5−5, 8−8
overload resolution and 13−9
`#else` 16−3
`else` 6−2
empty
argument list 8−8
class `sizeof` 9−1
statement 6−1
`empty` 20−4, 24−11
`basic_string` 21−14
`operator<` 20−4
`operator==` 20−4
`#endif` 16−3
`endl` 27−49, 51
end-of-file 23−13
`ends` 27−51
entity 3−1
enum name, `typedef` 7−6
`enum`
declaration `{}` 7−11
name, elaborated 7−9
overloading and 13−2
type of 7−11/12
type specifier 7−9
enumerated type 3−14, 17−5
enumeration 7−11
constant 7−11
example 7−12
linkage of 3−8
type, conversion to 5−9
type, `static_cast`, conversion to 5−9
underlying type 7−12
enumerator
class, scope of 7−13
definition 3−3
member 7−13
point of declaration 3−6
point of definition 7−12
redefinition 7−12
restriction 7−12
value of 7−11
enumerator 7−11
environment, program 3−9
`eof`
`ios_base` 27−17
`ios_traits` 27−9
`eos`
`ios_traits` 27−10
`string` 21−22
`string_char_traits` 21−4, 13/14
`wstring` 21−23
`epptr`, `basic_streambuf` 27−29
`eq`
`string` 21−22
`string_char_traits` 21−4, 15/17

wstring 21–23
eq_char_type, ios_traits 27–9
eq_int_type, ios_traits 27–9
equal 25–11
istreambuf_iterator 24–24
ostreambuf_iterator 24–27
equality operator 5–20
equality-expression 5–20
equal_range 25–21
equal_to 20–7
equivalence
template type 14–19
type 7–5, 9–2
equivalent
parameter declarations 13–2
parameter declarations, overloading and 13–2
erase
deque 23–16
list 23–18
vector 23–25
<errno.h> C–12
error
checking, point of 14–4
directive 16–8
#error 16–8
escape
character —see backslash
sequence 2–7
sequence, undefined 2–7
escape-sequence 2–7
evaluation
difference from C expression C–1
new, unspecified order of 5–16
of default argument 8–11/12
of expression, order of 5–1
order of argument 5–5
unspecified order of 3–10, 5–1
unspecified order of argument 5–5
unspecified order of function call 5–5
example
*const 8–4
array 8–7
class definition 9–4
const 8–4
constant pointer 8–4
constructor 12–2
constructor and initialization 12–9
declaration 3–2, 8–9
declarator 8–2
definition 3–2
delete 12–8
derived class 10–1
destructor and delete 12–9
ellipsis 8–10
enumeration 7–12
explicit destructor call 12–7
explicit qualification 10–4
friend 9–2
friend function 11–4
function declaration 8–10
function definition 8–13
linkage consistency 7–3
local class 9–13
member function 9–7, 11–4
member name access 11–3
nested class 9–11
nested class definition 9–12
nested class forward declaration 9–12
nested type name 9–13
of default argument 8–10/11
of incomplete type 3–13
of overloading 13–1
pointer to member 8–6
pure virtual function 10–9
scope of delete 12–9
scope resolution operator 10–4
static member 9–10
subscripting 8–7
type name 8–2
typedef 7–5
unnamed parameter 8–13
variable parameter list 8–10
virtual function 10–6/7
exception
allowing an 15–6
and new 5–16
arithmetic 5–1
declaration scope 3–4
explicit destructor and 12–7
handler 15–3, 17–21
handler, incomplete@type@in 15–3
handling 15–1
handling constructor 15–3
handling destructor 15–3
implementation dependency arithmetic 5–1
throwing 15–1
<exception> 17–18, 18–17
exception
exception 19–2/3, 5
exception 19–2/3, 5
what 19–2
~exception 19–2
~exception, exception 19–2
exception-declaration 15–1
exceptions, ios_base 27–17
exception-specification 15–5
exit from scope, destructor and 6–4
exit 3–10, 6–5, 18–11, 14
exp 26–21
complex 26–9
explanation, subscripting 8–7
explicit
constructor call 12–2
destructor and exception 12–7
destructor call 12–7
destructor call example 12–7
instantiation syntax 14–14
qualification 3–6, 7–14
qualification example 10–4
type conversion —see casting
explicit specifier 7–5
exponent-part 2–8
expression 5–1
ambiguity, declaration versus 6–6
assignment 5–22
constant 5–23
evaluation, difference from C C–1
order of evaluation of 5–1
parenthesized 5–3
pointer to member constant 5–12
postfix 5–4
primary 5–3
reference 5–1
statement 6–1
unary 5–12
unspecified 5–5
expression 5–22
expression-list 5–4
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" 17−19/C−14
"C++" C−13/14
"C++" 17−19
declaration 3−2
linkage of 7−3
linkage specification 7−23
reference declaration 8−17
restriction 7−3
external linkage 3−8, 17−19/C−14

F

F suffix 2−8
f suffix 2−8
facet, locale 22−4
fail, ios_base 27−17
failure, ios_base::failure 27−13
fclose 27−76
field declaration : 9−11
file 2−1
scope 17−19
source 2−1, 17−18/19
filebuf 27−73
fill 25−15
basic_ios 27−18
gslice_array 26−27
indirect_array 26−31
mask_array 26−29
slice_array 26−24
valarray 26−21
fill_n 25−15
final overrider 10−6
find 25−9
basic_string 21−15
find_end 25−10
find_first_not_of, basic_string 21−16
find_first_of 25−10
basic_string 21−15
find_if 25−9
find_last_not_of, basic_string 21−17
find_last_of, basic_string 21−16
floctal-digit 2−6
fixed 27−24
flags, ios_base 27−18
float
literal 2−8
type 3−15
type specifier 7−8
<float.h> 2−5/C−12
floating
point conversion 4−3
point conversion, implementation dependency 4−3
point conversion, undefined 4−3
point literal 2−8
point literal, type of 2−8
point promotion 4−3
point type 3−14
point type 3−15
point type, implementation dependency 3−15
floating-constant 2−8
floating-integral conversion 4−3
floating-suffix 2−8
float_round_style 18−8
flush 27−16, 36, 46, 52
basic_ostream 27−46
fmtflags
ios 27−52
ios_base 27−13
fopen 27−75/76
for
scope of declaration in 6−4
statement 6−3/4
statement, continue in 6−4
statement, declaration in 6−4
for_each 25−9
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 9−12
forward_iterator 24−11
distance_type 24−13
iterator_category 24−12
value_type 24−12
forward_iterator_tag 24−11
fprintf 27−47
fractional-constant 2−8
free
store —see also new, delete
store and constructor anachronism C−11
store and destructor anachronism C−11
free 20−18
valarray 26−22
freestanding implementation 17−18
friend
class, scope of 11−5
declaration, scope of 3−5
function, scope of 11−5
specifier 17−20
friend
access specifier and 11−5
class 11−5
class access and 11−5
declaration, overloaded name and 11−4
example 9−2
function, access and 11−4
function example 11−4
function, inline 11−5
function, linkage of 11−5
function, member function and 11−4
function, nested class 9−13
inheritance and 11−5
member function 11−4
specifier 7−6

template and 14–26
virtual and 10–7
front_inserter 24–19/20
front_insert_iterator 24–19
front_insert_iterator 24–20
front_insert_iterator 24–20
operator* 24–20
operator++ 24–20
operator= 24–20
fscanf 27–38
fseek 27–76
<fstream> 17–8, 27–72
full-expression 1–4
function
—see also friend function, member function, inline function, virtual function
allocation 3–12, 5–15, 12–8
argument —see argument
body 8–13
call 5–5
call evaluation, unspecified order of 5–5
call operator 5–4, 13–16
call operator, overloaded 13–17
call, recursive 5–5
call, undefined 5–10/11
cast, implementation dependency pointer to 5–10
cast, pointer to 5–10
comparison 17–1
comparison, pointer to 5–19
conversion, pointer to 4–2
deallocation 3–13, 5–17, 12–8
declaration 3–2, 8–8
declaration ambiguity 8–14
declaration, ellipsis in 5–5, 8–8
declaration example 8–10
declaration matching, overloaded 13–3
declarator () 8–8
definition 8–13
definition 3–3
definition anachronism, C C–10
definition anachronism, old style C–10
definition example 8–13
generated 14–9
global 17–19/20
handler 17–1
linkage specification 7–23
linkage specification overloaded 7–24
modifier 17–2
name hiding 13–3
name, overloaded 13–1
observer 17–2
operator 13–16
overloaded —see also overloading
parameter —see parameter
parameter adjustment 8–9
pointer to member 5–17
prototype scope 3–4
replacement 17–2
reserved 17–2
return —see return
return type —see return type
scope 3–4
scope of friend 11–5
specialized 14–9
specifier 7–4
template 14–19
template declaration 14–24
template definition 14–24
type 3–15, 8–8/9
viable 13–5
virtual —see virtual function
virtual member 17–7, 19/20
<functional> 20–5
function-body 8–13
function-definition 8–13
function-like macro 16–4
function-specifier 7–4
function-to-pointer conversion 4–2
fundamental
type 3–14
type conversion —see conversion, user-defined conversion
type, destructor and 12–7
fvoid_t 18–2

G

gbump, basic_streambuf 27–29
gcount, basic_istream 27–44
generate 25–15
generated
class 14–9
destructor —see default destructor
function 14–9
generate_n 25–15
get, basic_istream 27–41
getline, basic_istream 27–42
getloc, ios_base 27–19
get_off, ios_conv_traits<STATE_T> 27–65
get_offstate, ios_conv_traits<STATE_T> 27–66
get_offupos, ios_conv_traits<STATE_T> 27–66
get_pos, ios_conv_traits<STATE_T> 27–65
get_posstate, ios_conv_traits<STATE_T> 27–66
get_posupos, ios_conv_traits<STATE_T> 27–66
get_temporary_buffer 20–16
global
anonymous union 9–11
function 17–19/20
name 3–5
namespace scope 3–5
scope 3–5
global, locale 22–7
good, ios_base 27–17
goto
initialization and 6–5
statement 6–1, 4/5
gptr, basic_streambuf 27–29
grammar A–1
greater
than operator 5–19
than or equal to operator 5–19
greater 20–8
greater_equal 20–8
gslice
class 26–24
gslice 26–26
gslice 26–26
length 26–26
start 26–26
stride 26–26
gslice_array 26–26
fill 26–27
gslice_array 26–27
gslice_array 26–27
operator%= 26–27
operator&= 26–27
operator*= 26–27
operator+= 26–27
operator-= 26–27
operator/= 26–27

operator<<= 26−27
operator= 26−27
operator>>= 26−27
operator^= 26−27
operator|= 26−27

H

handler
exception 15−3, 17−21
function 17−1
incomplete@type@in exception 15−3
handler 15−1
handler-seq 15−1
has, locale 22−6
hash, collate 22−19
header
C 17−7, 19, 18−2/C−14
C++ 17−17/C−12
headers
ISO C 2−5
library 2−5
standard 2−5
hex number 2−8
hex 27−23
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
hosted implementation 17−18

I

id, qualified 5−3
id, locale 22−4
identifier 2−4, 5−3, 7−1
_, underscore in 2−5
identifier 2−4
identities and overloading, operator 13−16
id-expression 5−3
id-expression 5−3
#if 16−2, 17−19
if statement 6−2
#ifdef 16−3
if-else ambiguity 6−2
#ifndef 16−3
ifstream 27−73
ignore, basic_istream 27−43
imag 26−7
complex 26−9
imbue, ios_base 27−19
implementation
defined pointer integer conversion 5−10
dependency __STDC__ 16−9
dependency alignment of bit-field 9−11
dependency alignment requirement 3−13
dependency allocation 9−4, 11−2
dependency arithmetic exception 5−1
dependency asm 7−23
dependency bit-field allocation 9−11
dependency constructor and new 5−16
dependency division 5−18
dependency extension to C C−10
dependency floating point conversion 4−3
dependency floating point type 3−15
dependency generation of temporary 12−2
dependency linkage of main() 3−10
dependency linkage specification 7−23
dependency modulus 5−18
dependency object linkage 7−24
dependency overflow 5−1
dependency parameters to main() 3−9
dependency pointer comparison 5−20
dependency pointer subtraction 5−19
dependency pointer to function cast 5−10
dependency range of types 2−5
dependency right shift 5−19
dependency sign of bit-field 9−11
dependency sign of char 3−14
dependency sizeof expression 5−14
dependency sizeof integral type 3−14
dependency sizeof type 3−14
dependency string literal 2−8
dependency type of integer literal 2−6
dependency type of ptrdiff_t 5−19
dependency type of size_t 5−14
dependency type of sizeof expression 2−5
dependency value of char literal 2−8
dependency value of multicharacter literal 2−7
dependency volatile 7−8
dependency wchar_t 3−15
freestanding 17−18
hosted 17−18
limits 1−2
implementation-defined 4−3, 8−18, 17−18/19, 18−2/3, 11/12, 14, 19−2, 27−3, 17, 27−67/C−13
behavior 1−2
behavior reentrancy 17−20
message 27−13
smanip 27−52
streamoff 27−7/8
implementation-dependent 27−36, 46
implementation-generated definitions 3−2
implemetation-defined types 17−5
implicit
conversion 4−1, 5−1, 12−3
conversion sequence 4−1
conversion sequences implied object parameter 13−5
destructor call 12−7
user-defined conversion 12−5
implicitly-declared
copy assignment operator 12−17
copy constructor 12−16
default constructor 12−1
default constructor —see default constructor
implict object argument 13−5
implied
object parameter 13−5
object parameter, implicit conversion sequences 13−5
in_avail, basic_streambuf 27−27
#include 16−3, 17−18/19
includes 25−23
inclusion
conditional 16−2
source file 16−3
incomplete
class, cast to 5−10
type 3−13
type, example of 3−13
incompletely-defined object type 3−13
incomplete@type@in exception handler 15−3
increment
bool 5−7, 13
operator 5−7, 12/13
operator, overloaded 13−17
indeterminate uninitialized variable 8−14
indirect base class 10−1
indirect_array 26−29
fill 26−31

indirect_array 26−30
indirect_array 26−30
operator%= 26−30
operator&= 26−30
operator*= 26−30
operator+= 26−30
operator−= 26−30
operator/= 26−30
operator<<= 26−30
operator= 26−30
operator>>= 26−30
operator^= 26−30
operator|= 26−30
indirection 5−12
operator 5−12
inequality operator 5−20
inheritance 10−1
—see also multiple inheritance
and friend 11−5
of constructor 12−1
of destructor 12−6
of overloaded operator 13−16
of user-defined conversion 12−5
Init, ios_base::Init 27−16
init
basic_ios 27−36, 46
ios_base 27−20
~Init, ios_base::Init 27−16
init-declarator 8−1
init-declarator-list 8−1
initialization 8−13
and goto 6−5
and new 5−15
array 8−15
array of class objects 8−16, 12−10
auto 6−6
auto object 8−14
automatic 6−5/6
base class 12−10
character array 8−16
class member 8−14
class object 8−15, 12−9
class object —see also constructor
const 7−7, 8−15
const member 12−10
constructor and 12−9
default 8−14
default constructor and 12−9
definition and 7−2
dynamic 3−10
example, constructor and 12−9
extension to C memberwise C−2
in block 6−5
jump past 6−3, 5
local static 6−6
member 12−10
member object 12−10
order of 3−10, 10−2
order of base class 12−11
order of member 12−11
order of virtual base class 12−11
overloaded assignment and 12−10
parameter 5−5
reference 8−6, 17
reference member 12−10
run-time 3−10
static member 9−10
static object 3−10, 8−14
struct 8−15
union 8−16, 9−11
virtual base class 12−11, 17
initializer 8−14
base class 8−13
constant 9−3
list { } 8−15
member 8−13
scope of member 12−12
temporary and declarator 12−3
initializer 8−13
initializer-clause 8−14
initializer-list 8−14
injection from template, name 14−9
inline 17−19
friend function 11−5
function 7−4
function, extension to C C−1
function, linkage of 7−3
member function 9−6
inline
linkage of 3−8
specifier 7−4
inner_product 26−32
inplace_merge 25−23
input_iterator 24−11
distance_type 24−13
iterator_category 24−12
value_type 24−12
input_iterator_tag 24−11
insert
basic_string 21−11
deque 23−15
list 23−18
vector 23−24
inserter 24−21
insert_iterator 24−20
insert_iterator 24−21
insert_iterator 24−21
operator* 24−21
operator++ 24−21
operator= 24−21
instantiation
and specialization 14−12
multiple 14−12
point of 14−9
syntax, explicit 14−14
template 14−9
int, bool promotion to 4−2
int
type 3−14
type specifier 7−8
type, unsigned 3−14
integer
cast, pointer to 5−10
constant 2−6
conversion 4−3
conversion, implementation defined pointer 5−10
conversion, signed unsigned 4−3
literal 2−6
literal, base of 2−6
literal, implementation dependency type of 2−6
literal, type of 2−6
to pointer cast 5−10
type 3−15
integer-literal 2−5
integer-suffix 2−6
integral
promotion 4−2, 5−5
type 3−14
type 3−15
type, implementation dependency sizeof 3−14
value, undefined unrepresentable 4−3
internal linkage 3−8

internal 27−23
interpretation
 of binary operator 13−17
 of unary operator 13−16
INT_T 27−2
invalid_argument 19−3, 23−9/10
 invalid_argument 19−4
 invalid_argument 19−4
invocation, macro 16−5
<iomanip> 27−33
<ios> 27−7
ios 27−4, 7
 fmtflags 27−52
ios_base 27−11
 bad 27−17
 clear 27−17
 copyfmt 27−16
 eof 27−17
 exceptions 27−17
 fail 27−17
 flags 27−18
 fmtflags 27−13
 getloc 27−19
 good 27−17
 imbue 27−19
 init 27−20
 ios_base 27−20
 ios_base 27−20
 iostate 27−14
 iword 27−19
 openmode 27−15
 operator bool 27−16
 operator! 27−16
 precision 27−18
 pword 27−19
 rdstate 27−17
 seekdir 27−15
 setf 27−18
 setstate 27−17
 unsetf 27−18
 width 27−19
 xalloc 27−19
ios_base::failure 27−13
 failure 27−13
ios_base::Init 27−15
 Init 27−16
 ~Init 27−16
ios_conv_traits 27−63
ios_conv_traits<STATE_T> 27−63
 convin 27−64
 convout 27−65
 get_off 27−65
 get_offstate 27−66
 get_offupos 27−66
 get_pos 27−65
 get_posstate 27−66
 get_posupos 27−66
ios_conv_traits<wstreampos> 27−66
<iosfwd> 27−4
iostate, ios_base 27−14
ios_traits 27−8
 copy 27−10
 eof 27−9
 eos 27−10
 eq_char_type 27−9
 eq_int_type 27−9
 is_eof 27−10
 is_whitespace 27−10
 length 27−10
 newline 27−10
 not_eof 27−10
 to_char_type 27−9
 to_int_type 27−9
ios_traits<char> 27−7, 9
ios_traits<wchar_t> 27−7, 9
<iostream> 27−4
ipfx, basic_istream 27−36
is
 ctype 22−11
 ctype<char> 22−14
isalnum 22−8
isalpha 22−8
iscntrl 22−8
is_del, string_char_traits 21−4
isdigit 22−8
is_eof, ios_traits 27−10
isfx, basic_istream 27−37
isgraph 22−9
islower 22−8
ISO
 C headers 2−5
 C summary, compatibility with C−2
<iso646.h> C−12/13
is_open
 basic_filebuf 27−75
 basic_ifstream 27−80
 basic_ofstream 27−81
isprint 22−8
ispunct 22−8
isspace 22−7
<istream> 27−33
istream 27−4, 33
istreambuf_iterator 24−23
 equal 24−24
 istreambuf_iterator 24−24
 istreambuf_iterator 24−24
 iterator_category 24−25
 operator!= 24−25
 operator* 24−24
 operator++ 24−24
 operator== 24−25
 proxy 24−23
istream_iterator 24−21
 operator== 24−22
istringstream 27−54
isupper 22−8
is_whitespace, ios_traits 27−10
isxdigit 22−8
iteration statement 6−3
iteration-statement 6−3, 5
 scope 6−3
Iterator Requirements 24−4
<iterator> 24−1
iterator_category
 T* 24−12
 bidirectional_iterator 24−12
 forward_iterator 24−12
 input_iterator 24−12
 istreambuf_iterator 24−25
 operator!= 24−27
 ostreambuf_iterator 24−27
 output_iterator 24−12
 random_access_iterator 24−12
iword, ios_base 27−19

J

Jessie 12−3
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 3–13

L

`L`
 prefix 2–7, 9
 suffix 2–6, 8
`l` suffix 2–6, 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, sub-object
layout
 access specifier and object 11–2
 bit-field 9–11
 class object 9–4, 10–2
layout-compatible type 3–14
left
 shift operator 5–19
 shift, undefined 5–19
`left` 27–23
length of name 2–4
`length`
 `bits` 23–12
 `gslice` 26–26
 `ios_traits` 27–10
 `slice` 26–23
 `string` 21–22
 `string_char_traits` 21–5, 9/12, 15/19
 `valarray` 26–16
 `wstring` 21–23
`length_error` 19–4, 21–7
 `length_error` 19–4
 `length_error` 19–4
less
 than operator 5–19
 than or equal to operator 5–19
`less` 20–8
`less_equal` 20–8
lexical conventions 2–1
`lexicographical_compare` 25–28
library
 C++ Standard 17–1, 19/21
 Standard C 17–1, 6, 17–17/C–12, C–14
 Standard C++ C–12
 headers 2–5
limits, implementation 1–2
`<limits>` 18–3
`<limits.h>` 2–5/C–12
`#line` 16–8
linkage 3–1/2, 8
 consistency 7–3
 consistency example 7–3
 external 3–8, 17–19/C–14
 implementation dependency object 7–24
 internal 3–8
 of class 3–8
 of `const` 3–8, 7–3
 of enumeration 3–8
 of `extern` 7–3
 of `friend` function 11–5
 of `inline` 3–8
 of inline function 7–3
 of `main()`, implementation dependency 3–10
 of `static` 3–8, 7–3, 24
 specification 7–23
 specification class 7–23
 specification consistency 7–23
 specification, `extern` 7–23
 specification function 7–23
 specification, implementation dependency 7–23
 specification object 7–24
 specification overloaded function 7–24
 template 14–9
 to C 7–23
linkage-specification 7–23
list
 keyword 2–4
 operator 2–5, 13–16
 `{}`, initializer 8–15
`<list>` 23–7
`list` 23–16
 `assign` 23–18
 `erase` 23–18
 `insert` 23–18
 `merge` 23–19
 `remove` 23–19
 `resize` 23–18
 `reverse` 23–19
 `sort` 23–20
 `splice` 23–19
 `unique` 23–19
literal 2–5, 5–3
 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 2–8
 implementation dependency type of integer 2–6
 implementation dependency value of `char` 2–8
 implementation dependency value of multicharacter 2–7
 integer 2–6
 `long` 2–6
 `long double` 2–8
 multicharacter 2–7
 octal 2–6
 type of character 2–7
 type of floating point 2–8
 type of integer 2–6
 `unsigned` 2–6
literal 2–5
local
 class definition 9–13
 class example 9–13
 class member function 9–13
 class, member function in 9–7
 class restriction 9–13
 class restriction, `static` member 9–10
 class, scope of 9–13
 object, `static` 3–11
 object storage duration 3–12
 scope 3–4
 `static`, destruction of 6–6
 `static` initialization 6–6
 variable, destruction of 6–4/5
`<locale>` 22–1
`locale`

category 22–4
classic 22–7
facet 22–4
global 22–7
has 22–6
id 22–4
locale 22–4
locale 22–4
name 22–6
operator!= 22–6
operator() 22–7
operator== 22–6
transparent 22–7
use 22–6
<locale.h> C–12/13
locale-specific behavior 1–2
log 26–21
complex 26–9
log10 26–21
logical
AND operator 5–21
AND operator, side effects and 5–21
OR operator 5–21
OR operator, side effects and 5–21
negation operator 5–12/13
logical_and 20–8
logical_not 20–8
logical_or 20–8
logic_error 18–17, 19–3
logic_error 18–17, 19–3/4
logic_error 18–17, 19–3/4
long
constant 2–6
double literal 2–8
double type 3–15
literal 2–6
type 3–14
type specifier 7–8
type, unsigned 3–14
typedef and 7–2
longjmp 18–20
long-suffix 2–6
lookup
member name 10–3
name 3–1
template name 14–3
lower_bound 25–21
lowercase 17–6
lt
string 21–22
string_char_traits 21–4
wstring 21–23
lvalue 3–17
assignment and 5–22
cast 5–9/10
cast, reinterpret_cast, 5–10
cast, static_cast, 5–9
conversion to rvalue 4–1
modifiable 3–17
lvalue-to-rvalue conversion 4–1

M

macro
definition scope 16–6
function-like 16–4
invocation 16–5
masking 17–19
name 16–5
object-like 16–4
parameters 16–5
preprocessor 16–1
replacement 16–4
main() 3–9
implementation dependency linkage of 3–10
implementation dependency parameters to 3–9
parameters to 3–9
return from 3–10
make_heap 25–26
make_pair 20–4
malloc 20–18
management anachronism, memory C–11
<map> 23–26
map 23–27
operator< 23–29
operator== 23–29
operator[] 23–29
mask_array 26–27
fill 26–29
mask_array 26–28
mask_array 26–28
operator%= 26–29
operator&= 26–29
operator*= 26–29
operator+= 26–29
operator-= 26–29
operator/= 26–29
operator<<= 26–29
operator= 26–28
operator>>= 26–29
operator^= 26–29
operator|= 26–29
masking macro 17–19
<math.h> C–12
max 25–27
valarray 26–21
max_element 25–27
max_size, basic_string 21–13
meaning of declarator 8–4
member
—see also base class member
access operator, overloaded 13–17
access ambiguity 10–3
access, base class 10–1
access, class 5–6
access, protected 11–6
access, struct default 9–1
access, union default 9–1
array 9–4
cast, pointer to 5–10
class object 9–4
constructor order of execution 12–2
declaration 9–3
declaration, class 9–3
declaration, static 3–2
definition, static 9–10
destructor order of execution 12–6
enumerator 7–13
example, static 9–10
function and access control 12–1
function and friend function 11–4
function call, undefined 9–7
function, class 9–6
function, const 9–8
function, constructor and 12–2
function definition 9–6
function, destructor and 12–6
function example 9–7, 11–4
function, friend 11–4
function in local class 9–7
function, inline 9–6

function, local class 9−13
function, nested class 9−12
function, overload resolution and 13−5
function, `static` 9−9
function template 14−26
function, `union` 9−10
function, virtual 17−7, 19/20
function, `volatile` 9−8
initialization 12−10
initialization, `const` 12−10
initialization, order of 12−11
initialization, reference 12−10
initialization, `static` 9−10
initializer 8−13
initializer, scope of 12−12
local class restriction, `static` 9−10
name access 11−1
name access example 11−3
name lookup 10−3
name, overloaded 9−4
object initialization 12−10
of class type restriction 12−11
pointer to —see pointer to member
pointer value, null 4−4
`static` 9−9
`static` class 3−11
storage duration, class 3−13
template and `static` 14−27
type of `static` 5−12, 9−9
use, `static` 9−9
member-declaration 9−3
member-declarator 9−3
member-specification 9−3
memberwise
assignment, extension to C C−2
initialization, extension to C C−2
`memchr` 21−26
mem-initializer 12−10
memory
management —see also `new`, `delete`
management anachronism C−11
model 1−3
`<memory>` 20−11
`merge` 25−22
`list` 23−19
message, diagnostic 1−2
`message`, implementation-defined 27−13
`messages` 22−25
`messages_byname` 22−26
`min` 25−27
`valarray` 26−21
`min_element` 25−27
`minus` 20−7
`mismatch` 25−11
missing storage class specifier 7−3
modifiable lvalue 3−17
modifier function 17−2
modulus
implementation dependency 5−18
operator 5−18
`modulus` 20−7
`money_get` 22−23
`moneypunct` 22−24
`moneypunct_byname` 22−25
`money_put` 22−23
most derived class 12−11
multibyte
character 1−2
string, null-terminated 17−6
multicharacter
literal 2−7
literal, implementation dependency value of 2−7
multidimensional
array 8−8
array declarator 8−7
`multimap` 23−29
`operator<` 23−30
`operator==` 23−30
multiple
declaration 3−9
inheritance 10−1/2
inheritance DAG 10−3
inheritance, extension to C C−2
inheritance, `virtual` and 10−7
instantiation 14−12
multiplication operator 5−18
multiplicative operator 5−18
multiplicative-expression 5−18
`multiset` 23−32
`operator<` 23−33
`operator==` 23−33
mutable keyword 3−13

N

name 2−4, 3−1, 5−3
address of qualified 5−12
and translation unit 3−2
class —see class name
declaration 3−2
dependent 14−6
elaborated `enum` 7−9
global 3−5
hiding 3−5/6, 5−3, 6−5
hiding, class definition 9−2
hiding, function 13−3
hiding, overloading versus 13−3
hiding, user-defined conversion and 12−5
injection from template 14−9
length of 2−4
lookup 3−1
lookup, member 10−3
lookup, template 14−3
macro 16−5
overloaded function 13−1
overloaded member 9−4
point of declaration 3−6
qualified 7−14
reserved 17−19
resolution, template 14−3
scope of 3−4
space, label 6−1
type —see type name
unreserved 17−7
use of template 14−5
`name`
`locale` 22−6
`type_info` 18−16
namespace 17−8, 17−19/C−12
definition 7−13
scope 3−5
scope, global 3−5
namespaces 7−13
`narrow`
`ctype` 22−12
`ctype<char>` 22−15
`NDEBUG` 17−18
`ne`
`string` 21−22
`string_char_traits` 21−4
`wstring` 21−23

negate 20–7
negation operator, logical 5–12/13
nested
  class anachronism, scope of C–12
  class definition 9–11
  class definition example 9–12
  class example 9–11
  class forward declaration example 9–12
  class friend function 9–13
  class member function 9–12
  class, scope of 9–11
  type name 9–13
  type name example 9–13
  type name, scope of 9–13
<new> 17–18/19, 18–11
new 3–12, 5–14/15, 12–8
  array 5–15
  array of class objects and 5–15
  constructor and 5–15
  default constructor and 5–15
  exception and 5–16
  extension to C C–1
  extension to C overloading C–2
  implementation dependency constructor and 5–16
  initialization and 5–15
  operator 17–19, 18–12/13, 20–18
  placement syntax 5–15
  scoping and 5–14
  storage allocation 5–14
  type of 12–8
  unspecified order of evaluation 5–16
new[], operator 17–19, 18–13/14
new-declarator 5–14
new-expression 5–14
<new.h> 2–5
new_handler 3–12, 17–19, 18–14
new-initializer 5–14
new-line 2–7
newline, ios_traits 27–10
new-placement 5–14
new-type-id 5–14
next_permutation 25–28
noboolalpha 27–22
nondigit 2–4
none, bits 23–13
nonnested class anachronism C–12
non-trivial
  constructor 12–1
  destructor 12–6
nonvirtual base class DAG 10–3
nonzero-digit 2–6
norm, complex 26–9
noshowbase 27–22
noshowpoint 27–22
noshowpos 27–23
noskipws 27–23
not1 20–9
not2 20–9
notation, syntax 1–3
not_eof, ios_traits 27–10
not_equal_to 20–8
nouppercase 27–23
NPOS 18–2
NTBS 17–6, 27–47, 49, 52, 75
  static 17–6
nth_element 25–20
NTMBS 17–6, 18–15
  static 17–6, 18–15
NTWCS 17–7
  static 17–7
null
  character 0 2–9
  directive 16–9
  member pointer value 4–4
  pointer 5–19
  pointer constant 4–3/4
  pointer value 4–3
  reference 8–6
  statement 6–1
null-terminated
  byte string 17–6
  multibyte string 17–6
  wide-character string 17–7
number
  hex 2–8
  octal 2–8
<numeric> 26–31
numeric_limits 3–14, 18–3
num_get 22–15
numpunct 22–17
numpunct_byname 22–18
num_put 22–16

O

object 1–3, 3–1, 17
  class —see also class object
  complete 1–3
  definition 3–3
  delete 5–16
  destructor and placement of 12–7
  destructor static 3–10
  initialization, auto 8–14
  initialization, static 3–10, 8–14
  layout, access specifier and 11–2
  lifetime 3–11
  linkage, implementation dependency 7–24
  linkage specification 7–24
  static local 3–11
  storage duration, local 3–12
  temporary —see temporary
  type, completely-defined 3–13
  type, incompletely-defined 3–13
  undefined deleted 3–13, 5–16
  unnamed 12–2
object-expression 5–1
object-like macro 16–4
observer function 17–2
oct 27–24
octal
  literal 2–6
  number 2–8
octal-escape-sequence 2–7
octal-literal 2–6
of overloading, example 13–1
offsetof C–13
OFF_T 27–3
ofstream 27–73
old
  style base class initializer anachronism C–11
  style function definition anachronism C–10
one-definition rule 3–3
one's complement operator 5–12/13
open
  basic_filebuf 27–75
  basic_ifstream 27–80
  basic_ofstream 27–82
openmode, ios_base 27–15
operand
  const 5–1
  reference 5–1

volatile 5–1
operations on class object 9–1
operator
%= 5–22
&= 5–22
*= 5–22
+= 5–13, 22
–= 5–22
/= 5–22
<<= 5–22
>>= 5–22
^= 5–22
additive 5–18
address-of 5–12
assignment 5–22, 17–7
bitwise 5–20
bitwise AND 5–20
bitwise exclusive OR 5–20
bitwise inclusive OR 5–20
cast 5–12, 17, 8–2
class member access 5–6
comma 5–22
conditional expression 5–21
conversion 5–1, 12–4
copy assignment 12–16
decrement 5–7, 12/13
division 5–18
equality 5–20
example, scope resolution 10–4
function call 5–4, 13–16
function call 13–16
greater than 5–19
greater than or equal to 5–19
identities and overloading 13–16
increment 5–7, 12/13
indirection 5–12
inequality 5–20
left shift —see left shift operator
less than 5–19
less than or equal to 5–19
list 2–5, 13–16
logical AND 5–21
logical OR 5–21
logical negation 5–12/13
modulus 5–18
multiplication 5–18
multiplicative 5–18
new —see new
one's complement 5–12/13
overloaded 5–1
overloading —see also overloaded operator
overloading restrictions 13–16
pointer to member 5–17
precedence of 5–1
relational 5–19
right shift; right shift operator 5–19
scope resolution 3–5/6, 5–3, 9–6, 10–1, 8
shift —see left shift operator, right shift operator
side effects and comma 5–22
side effects and logical AND 5–21
side effects and logical OR 5–21
sizeof 5–12/13
subscripting 5–4, 13–16
unary 5–12
unary minus 5–12/13
unary plus 5–12/13
use, scope resolution 9–10
|= 5–22
operator T*, valarray 26–16
operator bool, ios_base 27–16
operator
delete 17–19, 18–12, 20–18
delete 5–17, 12–8
delete —see delete
delete[] 17–19, 18–13
delete[] 5–17, 12–8
function 13–16
new 17–19, 18–12/13, 20–18
new 5–15, 12–8
new[] 17–19, 18–13/14
new[] 5–15, 12–8
overloaded 13–16
operator!
ios_base 27–16
valarray 26–17
operator!= 20–3
basic_string 21–19
bits 23–12
complex 26–8
istreambuf_iterator 24–25
iterator_category 24–27
locale 22–6
ostreambuf_iterator 24–27
type_info 18–16
valarray 26–20
operator%, valarray 26–18/19
operator%=
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator&
bitmask 17–5
bits 23–13
valarray 26–18/19
operator&&, valarray 26–18/19
operator&=
bitmask 17–5
bits 23–10
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator(), collate_byname 22–19
operator(), locale 22–7
operator*
auto_ptr 20–18
back_insert_iterator 24–19
complex 26–7
front_insert_iterator 24–20
insert_iterator 24–21
istreambuf_iterator 24–24
ostreambuf_iterator 24–26
reverse_bidirectional_iterator 24–15
reverse_iterator 24–17
valarray 26–18/19
operator*=
complex 26–7
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator+
basic_string 21–18
complex 26–7
reverse_iterator 24–16
valarray 26–17/19
operator++
back_insert_iterator 24–19

front_insert_iterator 24–20
insert_iterator 24–21
istreambuf_iterator 24–24
ostreambuf_iterator 24–26
reverse_bidirectional_iterator 24–15
reverse_iterator 24–17
operator+=
basic_string 21–9
complex 26–6
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator–
complex 26–7
valarray 26–17/19
operator––
reverse_bidirectional_iterator 24–15
reverse_iterator 24–17
operator–=
complex 26–6
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator->, auto_ptr 20–18
operator/, valarray 26–18/19
operator/=
complex 26–7
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator< 26–21
basic_string 21–20
empty 20–4
map 23–29
multimap 23–30
multiset 23–33
pair 20–4
restrictor 20–5
set 23–31
valarray 26–20
vector 23–23
vector<bool> 23–26
operator<< 22–7, 27–23
basic_ostream 27–49
basic_string 21–21
bits 23–13/14
complex 26–9
valarray 26–18/19
operator<<=
bits 23–10
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator<= 20–3
basic_string 21–20
valarray 26–20
operator=
back_insert_iterator 24–19
basic_string 21–9
front_insert_iterator 24–20
gslice_array 26–27
indirect_array 26–30
insert_iterator 24–21
mask_array 26–28
ostreambuf_iterator 24–26
slice_array 26–24
type_info 18–16
valarray 26–16
operator==
basic_string 21–19
bits 23–12
complex 26–8
empty 20–4
istream_iterator 24–22
istreambuf_iterator 24–25
locale 22–6
map 23–29
multimap 23–30
multiset 23–33
ostreambuf_iterator 24–27
pair 20–4
queue 23–20
restrictor 20–5
reverse_bidirectional_iterator 24–15
reverse_iterator 24–18
set 23–31
stack 23–22
type_info 18–16
valarray 26–20
vector 23–23
vector<bool> 23–26
operator> 20–3, 26–21
basic_string 21–20
valarray 26–20
operator>= 20–3
basic_string 21–21
valarray 26–20
operator>> 22–7
basic_istream 27–39
basic_string 21–21
bits 23–13
complex 26–8
valarray 26–18/19
operator>>=
bits 23–10
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator[]
basic_string 21–12
map 23–29
valarray 26–16/17
operator^
bitmask 17–5
bits 23–13
valarray 26–18/19
operator^=
bitmask 17–5
bits 23–10
gslice_array 26–27
indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator|
bitmask 17–5
bits 23–13
valarray 26–18/19
operator|=
bitmask 17–5
bits 23–10
gslice_array 26–27

indirect_array 26–30
mask_array 26–29
slice_array 26–24
valarray 26–18/19
operator||, valarray 26–18/19
operator~
bitmask 17–5
bits 23–11
valarray 26–17
operator 13–16
operator-function-id 13–16
opfx, basic_ostream 27–46
optimization of temporary —see elimination of temporary
OR
operator, bitwise exclusive 5–20
operator, bitwise inclusive 5–20
operator, logical 5–21
operator, side effects and logical 5–21
order
of argument evaluation 5–5
of argument evaluation, unspecified 5–5
of base class initialization 12–11
of destruction of temporary 12–3
of evaluation new, unspecified 5–16
of evaluation of expression 5–1
of evaluation, unspecified 3–10, 5–1
of execution, base class constructor 12–2
of execution, base class destructor 12–6
of execution, constructor and array 12–9
of execution, constructor and static objects 12–10
of execution, destructor 12–6
of execution, destructor and array 12–6
of execution, member constructor 12–2
of execution, member destructor 12–6
of function call evaluation, unspecified 5–5
of initialization 3–10, 10–2
of member initialization 12–11
of virtual base class initialization 12–11
osfx, basic_ostream 27–46
<ostream> 27–33
ostream 27–4, 33
ostreambuf_iterator 24–25
equal 24–27
iterator_category 24–27
operator!= 24–27
operator* 24–26
operator++ 24–26
operator= 24–26
operator== 24–27
ostreambuf_iterator 24–26
ostreambuf_iterator 24–26
ostream_iterator 24–22
ostringstream 27–54
out_of_range 19–4, 21–7, 23–9/12
out_of_range 19–4
out_of_range 19–4
output_iterator 24–11
iterator_category 24–12
output_iterator_tag 24–11
overflow 5–1
implementation dependency 5–1
overflow
basic_convbuf 27–68
basic_filebuf 27–76
basic_streambuf 27–30
basic_stringbuf 27–57
overflow_error 19–5, 23–9, 11/12
overflow_error 19–6
overflow_error 19–6
overload
resolution 13–4
resolution and conversion 13–10
resolution and default argument 13–9
resolution and ellipsis 13–9
resolution and member function 13–5
resolution and pointer conversion 13–15
resolution contexts 13–4
resolution, template 14–23
overload keyword anachronism C–10
overloaded
assignment and initialization 12–10
assignment operator 13–17
binary operator 13–17
declaration 13–1
decrement operator 13–17
function, address of 5–13, 13–15
function ambiguity detection 13–4
function call operator 13–17
function call resolution —see also argument matching, overload resolution
function declaration matching 13–3
function, linkage specification 7–24
function name 13–1
increment operator 13–17
member access operator 13–17
member name 9–4
name and friend declaration 11–4
operator 13–16
operator 5–1
operator 13–16
operator and default argument 13–16
operator, inheritance of 13–16
subscripting operator 13–17
unary operator 13–16
overloading 8–9, 9–2, 13–1
and access 13–4
and const 13–1/2
and default argument 8–12
and default initializers 13–3
and delete 3–13
and derived class 13–3
and enum 13–2
and equivalent parameter declarations 13–2
and pointer 13–1
and pointer versus array 13–2
and reference 13–1
and return type 13–2
and scope 13–3
and specialization 14–25
and static 13–2
and typedef 13–2
and volatile 13–1/2
delete, extension to C C–2
extension to C C–1
new, extension to C C–2
operator identities and 13–16
postfix ++ and -- 13–17
prefix ++ and -- 13–17
resolution and access control 10–4
resolution, template function 14–19
restriction 13–16
subsequence rule 13–14
versus name hiding 13–3
overrider, final 10–6

P

pair 20–4
operator< 20–4
operator== 20–4
parameter 1–2, 8–13

adjustment, array 8–9
adjustment, function 8–9
declaration 8–8/9
example, unnamed 8–13
initialization 5–5
list example, variable 8–10
list, variable 5–5, 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()` 3–9
to `main()`, implementation dependency 3–9
parentheses
and ambiguity 5–14
in declaration 8–3/4
parenthesized expression 5–3
`partial_sort` 25–19
`partial_sort_copy` 25–20
`partial_sum` 26–32
`partition` 25–18
`pbackfail`
`basic_convbuf` 27–68
`basic_filebuf` 27–77
`basic_streambuf` 27–30
`basic_stringbuf` 27–57
`pbase`, `basic_streambuf` 27–29
`pbump`, `basic_streambuf` 27–29
`peek`, `basic_istream` 27–44
period 17–6
phases, translation 2–1
placement
of object, destructor and 12–7
syntax, `new` 5–15
`plus` 20–7
pm-expression 5–17
POD-struct 9–1
point
of declaration class name 9–3
of declaration enumerator 3–6
of declaration name 3–6
of definition, enumerator 7–12
of error checking 14–4
of instantiation 14–9
promotion, floating 4–3
type, floating 3–14
pointer
—see also `void*`
arithmetic 5–18
cast, integer to 5–10
comparison 5–19/20
comparison, implementation dependency 5–20
comparison, undefined 5–19/20
comparison, `void*` 5–19
constant, null 4–3/4
conversion 4–3
conversion, array 4–1
conversion, overload resolution and 13–15
declaration 8–4
declarator `*` 8–4
example, constant 8–4
integer conversion, implementation defined 5–10
null 5–19
overloading and 13–1
subtraction, implementation dependency 5–19
terminology 3–16
to abstract class 10–9
to bit-field restriction 9–11
to function cast 5–10
to function cast, implementation dependency 5–10
to function comparison 5–19
to function conversion 4–2
to integer cast 5–10
to member 3–15, 5–17
to member anachronism, cast of C–12
to member cast 5–10
to member constant expression 5–12
to member conversion 4–4
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 5–17
to member function, undefined bound C–12
to member operator 5–17
to member `void*` conversion 4–4
type 3–15
type extension to C, `void*` C–1
value, null 4–3
value, null member 4–4
versus array, overloading and 13–2
zero 4–3, 5–19
`pointer_to_binary_function` 20–11
`pointer_to_unary_function` 20–10
`polar`, `complex` 26–9
polymorphic
class 10–6
type 10–6
`pop`, `priority_queue` 23–21
`pop_heap` 25–26
`POS_T` 27–3
postfix
`++` and `--` 5–7
`++` and `--`, overloading 13–17
expression 5–4
potential scope 3–1
`pow` 26–21
`complex` 26–9
`pptr`, `basic_streambuf` 27–29
pragma directive 16–8
`#pragma` 16–8
precedence of operator 5–1
`precision`, `ios_base` 27–18
prefix
`++` and `--` 5–13
`++` and `--`, overloading 13–17
`L` 2–7, 9
preprocessing 16–1
directive 16–1
preprocessing-op-or-punc 2–5
preprocessing-token 2–2
preprocessor, macro 16–1
`prev_permutation` 25–28
primary expression 5–3
`priority_queue` 23–20
`pop` 23–21
`priority_queue` 23–21
`priority_queue` 23–21
`push` 23–21
`private` 11–1
base class 11–2
program 3–8
environment 3–9
start 3–9/10
startup 17–19
termination 3–10
termination and destructor 12–7
promotion

floating point 4–3
integral 4–2, 5–5
to int, bool 4–2
protected 11–1
extension to C C–2
member access 11–6
protection 17–20
—see access control
proxy, istreambuf_iterator 24–23
ptrdiff_t 5–19, 18–2
implementation dependency type of 5–19
ptr_fun 20–10/11
ptr-operator 8–1
public 11–1
base class 11–2
pubseekoff, basic_streambuf 27–27
pubseekpos, basic_streambuf 27–27
pubsetbuf, basic_streambuf 27–27
pubsync, basic_streambuf 27–27
punctuators 2–5
pure
specifier 9–3
virtual destructor 12–6
virtual function 10–9
virtual function call, undefined 10–9
virtual function definition 10–9
virtual function example 10–9
pure-specifier 9–3
push, priority_queue 23–21
push_heap 25–25
put, basic_ostream 27–51
putback, basic_istream 27–44
pword, ios_base 27–19

Q

qualification
conversions 4–2
explicit 3–6, 7–14
qualified
id 5–3
name 7–14
name, address of 5–12
qualified-id 5–3
question mark 2–7
<queue> 23–7
queue 23–20
operator== 23–20
quote
double 2–7
single 2–7

R

random_access_iterator 24–11
distance_type 24–13
iterator_category 24–12
value_type 24–12
random_access_iterator_tag 24–11
random_shuffle 25–18
range of types, implementation dependency 2–5
range_error 19–5
range\rror
range\rror 19–5
range\rror 19–5
rank, conversion 13–12
rdbuf
basic_ifstream 27–80
basic_ios 27–22
basic_istringstream 27–60
basic_ofstream 27–81
basic_ostringstream 27–61
rdstate, ios_base 27–17
reach 3–5
read, basic_istream 27–43
readsome, basic_istream 27–44
real 26–7
complex 26–10
realloc 20–18
recursive function call 5–5
redefinition
enumerator 7–12
typedef 7–5
reentrancy 17–20
implementation-defined behavior 17–20
reference 3–15
and argument passing 8–17
and return 8–17
argument 5–5
assignment 8–17
assignment to 5–22
binding 8–17
call by 5–5
cast 5–9, 11
cast, reinterpret_cast, 5–11
cast, static_cast, 5–9
const 8–18
declaration 8–5
declaration, extern 8–17
declarator & 8–5
expression 5–1
initialization 8–6, 17
member initialization 12–10
null 8–6
operand 5–1
overloading and 13–1
parameter 8–5
restriction 8–6
sizeof 5–13
type, extension to C C–1
reference-compatible 8–17
reference-related 8–17
region, declarative 3–1
register
declaration 7–3
restriction 7–3
reinterpret cast 5–10
reinterpret_cast
lvalue cast 5–10
reference cast 5–11
relational operator 5–19
relational-expression 5–19
release, auto_ptr 20–18
remainder operator —see modulus operator
remove 25–15
basic_string 21–11
list 23–19
remove_copy 25–16
remove_copy_if 25–16
remove_if 25–15
replace 25–14
basic_string 21–12
replace_copy 25–14
replace_copy_if 25–14
replace_if 25–14
replacement
function 17–2
macro 16–4
required behavior 17–2, 4
Requirements 17–3
Allocator 20–1

Container 23–1
Iterator 24–4
reraise 15–3
rescanning and replacement 16–6
reserve, vector 23–24
reserved
function 17–2
identifier 2–5
name 17–19
word —see keyword
reset
auto_ptr 20–18
bits 23–11
resetiosflags 27–52
resize
basic_string 21–13
deque 23–16
list 23–18
vector 23–24
resolution
and conversion, overload 13–10
and default argument, overload 13–9
and ellipsis, overload 13–9
and member function, overload 13–5
and pointer conversion, overload 13–15
argument matching —see overload
overload 13–4
overloaded function call resolution —see also argument matching, overload
overloading —see overload resolution
resolution overloading —see overload
scoping ambiguity 10–4
template function overloading 14–19
template name 14–3
template overload 14–23
restriction 17–19, 21
address of bit-field 9–11
anonymous union 9–11
auto 7–3
bit-field 9–11
constructor 12–1/2
copy assignment operator 12–18
copy constructor 12–17
destructor 12–6
enumerator 7–12
extern 7–3
local class 9–13
member of class type 12–11
overloading 13–16
pointer to bit-field 9–11
reference 8–6
register 7–3
static 7–3
static member local class 9–10
union 9–10, 12–2
restrictions, operator overloading 13–16
restrictor 20–4
operator< 20–5
operator== 20–5
rethrow 15–3
return
type 8–9
type conversion 6–5
type, overloading and 13–2
return 6–4/5
constructor and 6–5
from main() 3–10
reference and 8–17
statement —see also return
reverse 25–17
list 23–19
reverse_bidirectional_iterator 24–14
conversion 24–14
operator* 24–15
operator++ 24–15
operator-- 24–15
operator== 24–15
reverse_bidirectional_iterator 24–14
reverse_bidirectional_iterator 24–14
reverse_copy 25–17
reverse_iterator 24–15
conversion 24–17
operator* 24–17
operator+ 24–16
operator++ 24–17
operator-- 24–17
operator== 24–18
reverse_iterator 24–17
reverse_iterator 24–17
rfind, basic_string 21–15
right
shift, implementation dependency 5–19
shift operator 5–19
right 27–23
rotate 25–17
rotate_copy 25–17
rounding 4–3
rule, one-definition 3–3
rules
for *condition*s 6–2
summary, scope 3–6
run-time initialization 3–10
runtime_error 19–4
runtime_error 19–5/6
runtime_error 19–5/6
rvalue 3–17
lvalue conversion to 4–1

S

sbumpc, basic_streambuf 27–28
scan_is
ctype 22–12
ctype<char> 22–14
scan_not
ctype 22–12
ctype<char> 22–14
s-char 2–8
s-char-sequence 2–8
scientific 27–24
scope 3–1
class 3–5
destructor and exit from 6–4
exception declaration 3–4
file 17–19
function 3–4
function prototype 3–4
global 3–5
global namespace 3–5
iteration-statement 6–3
local 3–4
macro definition 16–6
namespace 3–5
of class definition 9–2
of class name 9–2
of declaration in for 6–4
of default argument 8–12
of delete example 12–9
of enumerator class 7–13
of friend class 11–5
of friend declaration 3–5

of friend function 11–5
of label 3–4, 6–1
of local class 9–13
of member initializer 12–12
of name 3–4
of nested class 9–11
of nested class anachronism C–12
of nested type name 9–13
of parameter 3–4
overloading and 13–3
potential 3–1
resolution operator 3–5/6, 5–3, 9–6, 10–1, 8
resolution operator `::` 3–6
resolution operator example 10–4
resolution operator use 9–10
rules summary 3–6
selection-statement 6–2
scoping
ambiguity resolution 10–4
and `new` 5–14
`search` 25–12
`seekdir`, `ios_base` 27–15
`seekoff`
`basic_convbuf` 27–69
`basic_filebuf` 27–78
`basic_streambuf` 27–32
`basic_stringbuf` 27–57
`seekpos`
`basic_convbuf` 27–70
`basic_filebuf` 27–78
`basic_streambuf` 27–32
`basic_stringbuf` 27–58
selection statement 6–2
selection-statement 6–2
scope 6–2
semantics, class member 5–6
separate
compilation 2–1
translation 2–1
sequence
implicit conversion 4–1
statement 6–1
sequencing operator —see comma operator
`<set>` 23–27
`set` 23–30
`bits` 23–11
`operator<` 23–31
`operator==` 23–31
`setbase` 27–53
`setbuf`
`basic_convbuf` 27–70
`basic_filebuf` 27–79
`basic_streambuf` 27–33, 59
`basic_stringbuf` 27–59
`set_difference` 25–24
`setf`, `ios_base` 27–18
`setfill` 27–53
`setg`, `basic_streambuf` 27–29
`set_intersection` 25–24
`setiosflags` 27–52
`setjmp` 17–19
`<setjmp.h>` C–12
`setlocale` 17–6
`set_new_handler` 17–19, 18–15
`setp`, `basic_streambuf` 27–30
`setprecision` 27–53
`setstate`, `ios_base` 27–17
`set_symmetric_difference` 25–25
`setterminate` 15–7
`set_terminate` 17–19, 18–18
`set_unexpected` 15–7, 17–19, 18–18
`set_union` 25–23
`setw` 27–53
`sgetc`, `basic_streambuf` 27–28
`sgetn`, `basic_streambuf` 27–28
shift operator —see left shift operator, right shift operator
`shift`, `valarray` 26–21
shift-expression 5–19
`short`
type 3–14
type specifier 7–8
type, `unsigned` 3–14
`typedef` and 7–2
`showbase` 27–22
`showmany`
`basic_filebuf` 27–77
`basic_streambuf` 27–31, 77
`showpoint` 27–22
`showpos` 27–22
side
effects 5–1
effects and comma operator 5–22
effects and logical AND operator 5–21
effects and logical OR operator 5–21
sign
of bit-field, implementation dependency 9–11
of `char`, implementation dependency 3–14
sign 2–8
`<signal.h>` C–12
signature 1–2
`signed`
`char` type 3–14
character 3–14
`typedef` and 7–2
`unsigned` integer conversion 4–3
simple-escape-sequence 2–7
simple-type-specifier 7–8
`sin` 26–21
`complex` 26–10
single
precision arithmetic, extension to C C–1
quote 2–7
`sinh`, `complex` 26–10
`sink` 26–21
`size`, `basic_string` 21–13
`sizeof`
array 5–13
class object 5–13
empty class 9–1
expression, implementation dependency 5–14
expression, implementation dependency type of 2–5
integral type, implementation dependency 3–14
operator 5–12/13
reference 5–13
string 2–9
type, implementation dependency 3–14
`size_t` 5–14, 18–2
implementation dependency type of 5–14
`skipws` 27–23
`slice` 26–22
`length` 26–23
`slice` 26–22
`slice` 26–22
`start` 26–23
`stride` 26–23
`slice_array` 26–23
`fill` 26–24
`operator%=` 26–24
`operator&=` 26–24
`operator*=` 26–24
`operator+=` 26–24
`operator-=` 26–24

operator/= 26–24
operator<<= 26–24
operator= 26–24
operator>>= 26–24
operator^= 26–24
operator|= 26–24
slize_array
slize_array 26–24
slize_array 26–24
smanip 27–52
implementation-defined 27–52
snextc, basic_streambuf 27–28
sort 25–19
list 23–20
sort_heap 25–26
source
file 2–1, 17–18/19
file inclusion 16–3
space, white 2–3
special member function —see also constructor, destructor, inline function, user-defined conversion, virtual function
specialization 14–9
instantiation and 14–12
overloading and 14–25
template 14–14
specialized
class 14–9
function 14–9
specification, template argument 14–19
specifier
access —see access specifier
auto 7–2
declaration 7–2
elaborated type 3–6
explicit 7–5
friend 7–6
friend 17–20
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
splice, list 23–19
sputbackc, basic_streambuf 27–28
sputc, basic_streambuf 27–28
sputn, basic_streambuf 27–28
sqrt 26–21
complex 26–10
<sstream> 27–53
stable_partition 25–18
stable_sort 25–19
stack unwinding 15–3
<stack> 23–7
stack 23–22
operator== 23–22
Standard
C library 17–1, 6, 17–17/C–12, C–14
C++ library C–12
library, C++ 17–1, 19/21
standard
conversion 4–1
headers 2–5
start, program 3–9/10
start
gslice 26–26
slice 26–23
startup, program 17–19
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 17–6
NTMBS 17–6, 18–15
NTWCS 17–7
cast 5–9
type 1–2
static
class member 3–11
data member 9–9
destruction of local 6–6
initialization, local 6–6
linkage of 3–8, 7–3, 24
local object 3–11
member 9–9
member declaration 3–2
member definition 9–10
member example 9–10
member function 9–9
member initialization 9–10
member local class restriction 9–10
member, template and 14–27
member, type of 5–12, 9–9
member use 9–9
object, destructor 3–10
object initialization 3–10, 8–14
objects order of execution, constructor and 12–10
overloading and 13–2
restriction 7–3
specifier 7–2
variable, template and 14–27
static_cast
conversion to enumeration type 5–9
lvalue cast 5–9
reference cast 5–9
<stdarg.h> C–12
__STDC__ 16–9
implementation dependency 16–9
<stddef> 18–1
<stddef.h> 2–5, 7, 2–9/C–13
<stdexcept> 19–1
<stdio.h> C–12/13
<stdlib.h> 2–5/C–13
storage
allocation new 5–14
class 3–1
class declaration 7–2

class specifier 7–2
class specifier, missing 7–3
duration 3–11
duration, auto 3–12
duration, class member 3–13
duration, dynamic 3–12, 5–14
duration, local object 3–12
management —see new, delete
of array 8–8
str
basic_istringstream 27–60
basic_ostringstream 27–62
basic_stringbuf 27–56
strchr 21–25
<streambuf> 27–24
streambuf 27–24
streamoff 27–7
implementation-defined 27–7/8
streamsize 27–8
stride
gslice 26–26
slice 26–23
string
concatenation 2–9
constant 2–8
distinct 2–8
literal 2–8
literal concatenation, undefined 2–9
literal, implementation dependency 2–8
literal, type of 2–8
literal, undefined change to 2–8
null-terminated byte 17–6
null-terminated multibyte 17–6
null-terminated wide-character 17–7
sizeof 2–9
terminator 0 2–9
type of 2–8
wide-character 2–9
<string> 21–1
string 21–21
assign 21–21
compare 21–22
copy 21–22
eos 21–22
eq 21–22
length 21–22
lt 21–22
ne 21–22
stringbuf 27–54
string_char_traits 21–3
assign 21–4
char_in 21–4
char_out 21–4
compare 21–5, 18
copy 21–5
eos 21–4, 13/14
eq 21–4, 15/17
is_del 21–4
length 21–5, 9/12, 15/19
lt 21–4
ne 21–4
string_char_traits<char> 21–21
string_char_traits<wchar_t> 21–22
<string.h> C–12/13
string-literal 2–8
strpbrk 21–25
strrchr 21–25
strstr 21–26
struct
class versus 9–1
default member access 9–1
initialization 8–15
type specifier 7–9
structure 9–1
tag —see class name
sub-object 1–3
lattice —see DAG
subscripting
example 8–7
explanation 8–7
operator 5–4, 13–16
operator, overloaded 13–17
subsequence rule, overloading 13–14
substr, basic_string 21–17
subtraction
implementation dependency pointer 5–19
operator 5–18
suffix
E 2–8
F 2–8
L 2–6, 8
U 2–6
f 2–8
l 2–6, 8
u 2–6
sum, valarray 26–20
summary
compatibility with C C–1
compatibility with ISO C C–2
scope rules 3–6
syntax A–1
sungetc, basic_streambuf 27–28
swap 25–13
basic_string 21–14
swap_ranges 25–13
switch
statement 6–2/3, 5
statement, declaration in 6–3
sync
basic_convbuf 27–70
basic_filebuf 27–79
basic_istream 27–37
basic_streambuf 27–33
synonym 7–17
type name as 7–5
syntax
checking 14–4
class member 5–6
explicit instantiation 14–14
notation 1–3
summary A–1

T

T*
distance_type 24–13
iterator_category 24–12
valarray operator 26–16
value_type 24–13
tan 26–21
tanh 26–21
template 14–1
access rules 14–18
and < 14–2
and friend 14–26
and static member 14–27
and static variable 14–27
argument 14–17
argument deduction 14–20
argument specification 14–19
class 14–2, 23–9

definition of 14–1
function 14–19
function overloading resolution 14–19
instantiation 14–9
linkage 14–9
member function 14–26
name injection from 14–9
name lookup 14–3
name resolution 14–3
name, use of 14–5
overload resolution 14–23
specialization 14–14
type equivalence 14–19
`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 14–15
template-parameter-list 14–1
temporary 12–2
and declarator initializer 12–3
constructor for 12–2
destruction of 12–2
destructor for 12–2
elimination of 12–2
implementation dependency generation of 12–2
order of destruction of 12–3
`terminate` 3–10, 15–7, 18–10, 18
`terminate()` 15–7
`terminate_handler` 17–19, 18–17
termination
and destructor, program 12–7
program 3–10
terminator `0`, string 2–9
terminology, pointer 3–16
`test`, `bits` 23–12
`this` 5–3
anachronism, assignment to C–11
and constructor anachronism C–11
and destructor anachronism C–11
pointer —see `this`
type of 9–8
`throw` 15–1
throw-expression in conditional-expression 5–21
throw-expression 15–1
throwing, exception 15–1
throw-point 15–1
`tie`, `basic_ios` 27–21
`time_get` 22–21
`time_get_byname` 22–21
`<time.h>` C–12/13
`time_put` 22–22
`time_put_byname` 22–22
`times` 20–7
to
int, bool promotion 4–2
rvalue, lvalue conversion 4–1
`to_char_type`, `ios_traits` 27–9
`toggle`, `bits` 23–11
`to_int_type`, `ios_traits` 27–9
token 2–3, 5
token 2–3
`tolower` 22–9
`ctype` 22–12
`ctype<char>` 22–15
`to_string`, `bits` 23–12
`to_ulong`, `bits` 23–12
`toupper` 22–9
`ctype` 22–12
`ctype<char>` 22–14
`to_ushort`, `bits` 23–11
`transform` 25–14
`collate` 22–19
translation
phases 2–1
separate 2–1
unit 17–18/19
unit 2–1, 3–8
unit, name and 3–2
translation@unit 2–1
`transparent`, `locale` 22–7
trigraph 2–1/2
truncation 4–3
`try` 15–1
try-block 15–1
type 3–1
Boolean 3–14
ambiguity, declaration 7–2
arithmetic 3–15
array 3–15, 8–9
bitmask 17–5
`char` 3–14
character 3–14
checking, argument 5–5
checking, extension to C C–1
checking of default argument 8–11
class and 9–1
completely-defined object 3–13
compound 3–15
`const` 7–7
conversion, argument 12–3
conversion, explicit —see casting
declaration 8–4
declaration consistency 3–9
declaration, `typedef` as 7–5
definition, class name as 9–2
destination 8–15
`double` 3–15
dynamic 1–2
enumerated 3–14, 17–5
enumeration underlying 7–12
equivalence 7–5, 9–2
equivalence, template 14–19
example of incomplete 3–13
extension to C reference C–1
extension to C user-defined C–1
`float` 3–15
floating point 3–14
function 3–15, 8–8/9
fundamental 3–14
generator —see template
implementation dependency `sizeof` 3–14
incomplete 3–13
incompletely-defined object 3–13
`int` 3–14
integral 3–14
`long` 3–14
`long double` 3–15
name 3–17, 8–2
name as synonym 7–5
name example 8–2
name example, nested 9–13
name, nested 9–13
name, scope of nested 9–13
of bit-field 9–11
of character literal 2–7
of constructor 12–2
of conversion 12–4
of `delete` 12–8
of `enum` 7–11/12

of floating point literal 2−8
of integer literal 2−6
of integer literal, implementation dependency 2−6
of new 12−8
of ptrdiff_t, implementation dependency 5−19
of size_t, implementation dependency 5−14
of sizeof expression, implementation dependency 2−5
of static member 5−12, 9−9
of string 2−8
of string literal 2−8
of this 9−8
pointer 3−15
polymorphic 10−6
short 3−14
signed char 3−14
specifier, char 7−8
specifier, class 7−9
specifier, double 7−8
specifier, elaborated 14−5
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 7−8
static 1−2
unsigned 3−14
unsigned char 3−14
unsigned int 3−14
unsigned long 3−14
unsigned short 3−14
void 3−15
void* 3−16
volatile 7−7
wchar_t 3−15
wchar_t underlying 3−15
typedef 3−17
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
overloading and 13−2
redefinition 7−5
specifier 7−5
typedef-name 7−5
typeid 5−8
type-id 8−2
type-id-list 15−5
<typeinfo> 17−18, 18−15
type_info 5−8, 18−15
before 18−16
name 18−16
operator!= 18−16
operator= 18−16
operator== 18−16
type_info 18−16
type_info 18−16
type-parameter 14−15
types
implementation dependency range of 2−5
implemetation-defined 17−5
type-specifier
bool 7−8
wchart 7−8
type-specifier 7−7

U

U suffix 2−6
u suffix 2−6
uflow, basic_streambuf 27−32
unary
expression 5−12
minus operator 5−12/13
operator 5−12
operator, interpretation of 13−16
operator, overloaded 13−16
plus operator 5−12/13
unary-expression 5−12
unary_function 20−6
unary_negate 20−9
unary-operator 5−12
#undef 16−6, 17−19
undefined 5−8, 17−2, 19, 18−20, 21−12, 22−4, 24−23, 25−20, 26−15, 17, 19/21, 25, 30, 27−2, 10
behavior 1−2
bound pointer to member function C−12
change to const object 7−7
change to string literal 2−8
delete 5−16
deleted object 3−13, 5−16
division by zero 5−1, 18
escape sequence 2−7
floating point conversion 4−3
function call 5−10/11
left shift 5−19
member function call 9−7
pointer comparison 5−19/20
pure virtual function call 10−9
string literal concatenation 2−9
unrepresentable integral value 4−3
value delete 5−16
underflow
basic_convbuf 27−68
basic_filebuf 27−78
basic_streambuf 27−31
basic_stringbuf 27−57
underlying
type, enumeration 7−12
type, wchar_t 3−15
underscore
character 17−19
character _ 2−4
in identifier _ 2−5
unexpected 15−7, 18−19
unexpected() 15−7
unexpected_handler 17−19, 18−18
unget, basic_istream 27−44
uninitialized variable, indeterminate 8−14
uninitialized_copy 20−17
uninitialized_fill 20−17
uninitialized_fill_n 20−17
union 3−15, 9−10
access control, anonymous 9−11
anonymous 9−10
class versus 9−1
constructor 9−10
default member access 9−1
destructor 9−10
extension to C anonymous C−1
global anonymous 9−11
initialization 8−16, 9−11

member function 9–10
restriction 9–10, 12–2
restriction, anonymous 9–11
type specifier 7–9
unique 25–16
list 23–19
unique_copy 25–16
unit, translation 17–18/19
unknown argument type 8–8
unnamed
bit-field 9–11
class 7–6
object 12–2
parameter example 8–13
unqualified-id 5–3
unrepresentable integral value, undefined 4–3
unreserved name 17–7
unsetf, ios_base 27–18
unsigned
arithmetic 3–15
char type 3–14
constant 2–6
int type 3–14
integer conversion, signed 4–3
literal 2–6
long type 3–14
short type 3–14
type 3–14
type specifier 7–8
typedef and 7–2
unsigned-suffix 2–6
unspecified 18–12/13, 15, 17, 21–8, 26–20, 27–9, 57, 68
argument to constructor 5–16
behavior 1–2
destructor call 6–6
expression 5–5
order of argument evaluation 5–5
order of evaluation 3–10, 5–1
order of evaluation new 5–16
order of function call evaluation 5–5
unwinding, stack 15–3
upper_bound 25–21
uppercase 17–6, 19
uppercase 27–23
use of template name 14–5
use, locale 22–6
user-defined
conversion 4–1, 5–1, 12–3/4
conversion and name hiding 12–5
conversion, implicit 12–5
conversion, inheritance of 12–5
conversion, virtual 12–5
type, extension to C C–1
using-declaration 7–18
using-directive 7–22
usual arithmetic conversions 5–2
<utility> 20–3

V

va_end 17–19
<valarray> 26–10
valarray 26–12, 26
apply 26–22
fill 26–21
free 26–22
length 26–16
max 26–21
min 26–21
operator T* 26–16
operator! 26–17
operator!= 26–20
operator% 26–18/19
operator%= 26–18/19
operator& 26–18/19
operator&& 26–18/19
operator&= 26–18/19
operator* 26–18/19
operator*= 26–18/19
operator+ 26–17/19
operator+= 26–18/19
operator- 26–17/19
operator-= 26–18/19
operator/ 26–18/19
operator/= 26–18/19
operator< 26–20
operator<< 26–18/19
operator<<= 26–18/19
operator<= 26–20
operator= 26–16
operator== 26–20
operator> 26–20
operator>= 26–20
operator>> 26–18/19
operator>>= 26–18/19
operator[] 26–16/17
operator^ 26–18/19
operator^= 26–18/19
operator| 26–18/19
operator|= 26–18/19
operator|| 26–18/19
operator~ 26–17
shift 26–21
sum 26–20
valarray 26–15
valarray 26–15
~valarray 26–15
~valarray, valarray 26–15
va_list 17–19
value
call by 5–5
delete, undefined 5–16
null member pointer 4–4
null pointer 4–3
of char literal, implementation dependency 2–8
of enumerator 7–11
of multicharacter literal, implementation dependency 2–7
undefined unrepresentable integral 4–3
value_type 24–12
T* 24–13
bidirectional_iterator 24–12
forward_iterator 24–12
input_iterator 24–12
random_access_iterator 24–12
variable
argument list 8–8
indeterminate uninitialized 8–14
parameter list 5–5, 8–8
parameter list example 8–10
template and static 14–27
<vector> 23–8
vector 23–22
assign 23–24
capacity 23–24
erase 23–25
insert 23–24
operator< 23–23
operator== 23–23
reserve 23–24
resize 23–24
vector 23–23

vector 23–23
vector<bool> 23–25
operator< 23–26
operator== 23–26
vertical tab 2–7
viable function 13–5
virtual
base class 10–2
base class DAG 10–3
base class assignment 12–18
base class dominance 10–5
base class initialization 12–11, 17
base class initialization, order of 12–11
destructor 12–6
destructor, pure 12–6
function 10–6
function access 11–7
function call 10–8
function call, constructor and 12–14
function call, destructor and 12–14
function call, undefined pure 10–9
function definition 10–7
function definition, pure 10–9
function example 10–6/7
function example, pure 10–9
function, pure 10–9
member function 17–7, 19/20
user-defined conversion 12–5
virtual
and friend 10–7
and multiple inheritance 10–7
specifier 7–5
visibility 3–5
void
parameter 8–8
type 3–15
type specifier 7–8
void& 8–5
void*
conversion, pointer to member 4–4
pointer comparison 5–19
pointer type extension to C C–1
type 3–16
volatile 3–16
constructor and 9–8, 12–1
destructor and 9–8, 12–6
extension to C C–2
implementation dependency 7–8
member function 9–8
operand 5–1
overloading and 13–1/2
type 7–7
type specifier 7–8

W

<wchar.h> C–12/13
wchart type-specifier 7–8
wchar_t 2–7, 9, 17–6, 18–3, 21–25, 27–83
implementation dependency 3–15
type 3–15
underlying type 3–15
<wctype.h> C–12
WEOF 18–3
werr 27–6
wfilebuf 27–73
what
bad_alloc 18–14
exception 19–2
while statement 6–3/4
white
space 2–3
space 2–3
wide-character 2–7
string 2–9
string, null-terminated 17–7
widen
ctype 22–12
ctype<char> 22–15
width, ios_base 27–19
wifstream 27–73
win 27–6
wint_t 18–2
wios 27–4, 7
wistream 27–33
wistringstream 27–54
wlog 27–6
wofstream 27–73
wostream 27–33
wostringstream 27–54
wout 27–6
write, basic_ostream 27–51
ws 27–39, 44
wstreambuf 27–24
wstreamoff 27–8
wstreampos 27–8
wstring 21–22
assign 21–23
compare 21–23
copy 21–23
eos 21–23
eq 21–23
length 21–23
lt 21–23
ne 21–23
wstringbuf 27–54

X

xalloc, ios_base 27–19
xsgetn, basic_streambuf 27–32
xsputn, basic_streambuf 27–32
XUNEXPECTED 18–18
X(X&) —see copy constructor , 16

Z

zero
pointer 4–3, 5–19
undefined division by 5–1, 18
width of bit-field 9–11