Index

, —see comma operator
! —see logical negation operator
!= —see inequality operator
operator 16−6
operator 16−6
% —see modulus operator
%= operator 5−27
&
 —see address-of operator
 —see bitwise AND operator
 reference declarator 8−6
&& —see logical AND operator
&= operator 5−27
()
 —see function call operator
 function declarator 8−9
*
 —see indirection operator
 —see multiplication operator
 pointer declarator 8−5
*= operator 5−27
+
 —see addition operator
 —see unary plus operator
++ —see increment operator
+= operator 5−15, 27
-
 —see subtraction operator
 —see unary minus operator
-- —see decrement operator
-= operator 5−27
-> —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 2−5
// comment 2−5
/= operator 5−27
:
 field declaration 9−11
 label specifier 6−1
::
 —see scope resolution operator
 scope resolution operator 3−13
::*, pointer to member declarator 8−7
<
 —see less than operator
 template and 14−4/5
<< —see left shift operator
<<= operator 5−27
<= —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 5−27
?: —see conditional expression operator
[]
 —see subscripting operator
 array declarator 8−8
\ —see backslash
^ —see bitwise exclusive OR operator
^= operator 5−27
_, underscore in identifier 2−6
{}
 block statement 6−1
 class declaration 9−1
 class definition 9−1
 enum declaration 7−10
 initializer list 8−18
| —see bitwise inclusive OR operator
|= operator 5−27
|| —see logical OR operator
~
 —see destructor
 —see one's complement operator
0
 —see also zero, null
 null character 2−11
 string terminator 2−11

A

abort 3−24, 6−5, 17−9, 18−9, 14, 19
abs 26−20, 32/33
 complex 26−8
abstract
 class 10−10
 class, constructor and 10−11
 class, pointer to 10−10
abstract-declarator 8−2
access
 adjusting base class member 11−4

ambiguity, member 10−4
and `friend`, class 11−6
and `friend` function 11−5
base class 11−3
base class member 10−1
class member 5−6
control 11−1
control, anonymous `union` 9−10
control default 11−1
control, member function and 12−1
control, overloading resolution and 10−5
declaration 11−4
example, member name 11−5
member name 11−1
overloading and 13−4
specifier 11−2/3
specifier and `friend` 11−7
specifier and object layout 11−3
`struct` default member 9−1
`union` default member 9−1
virtual function 11−9
access-specifier 10−1
`accumulate` 26−30
`acos` 26−20, 33
addition operator 5−22
additive operator 5−22
additive-expression 5−22
address
of bit-field 9−11
of bit-field restriction 9−11
of constructor 12−2
of cv-qualified name 5−14
of member function, unspecified 17−14
of overloaded function 5−14, 13−18
address-of operator 5−14
`adjacent_find` 25−11
adjusting base class member access 11−4
adjustment
array parameter 8−10
function parameter 8−10
`advance` 24−10
aggregate 8−18
alert 2−9
`<algorithm>` 25−1
alias 7−15
alignment
of bit-field 9−11
of bit-field, implementation defined 9−11
requirement, implementation-defined 3−31
allocation
function 3−26, 5−17, 12−9
implementation defined bit-field 9−11
`new`, storage 5−16
unspecified 9−5, 11−3
`Allocator` requirements 20−2
`allocator` 20−15
allowing an exception 15−7
altermate definition 17−11
`always_noconv`, `codecvt` 22−18
ambiguity
base class member 10−4
class conversion 10−6
declaration type 7−2
declaration versus cast 8−3
declaration versus expression 6−7
detection, overloaded function 13−4
function declaration 8−16
member access 10−4
parentheses and 5−16
resolution, scoping 10−5
Amendment 1 17−11
AND
operator, bitwise 5−25
operator, logical 5−25
operator, side effects and logical 5−25
and pointer to member type, multi-level mixed pointer 4−3
anonymous
`union` 9−10
`union` access control 9−10
`union` at namespace scope 9−10
`union`, global 9−10
`union` restriction 9−10
`any`, `bitset` 23−47
`append`, `basic_string` 21−17
`apply`, `valarray` 26−18
`arg`, `complex` 26−8
`argc` 3−22
argument 1−2, 17−13/14, 19−3
and name hiding, default 8−13
and virtual function, default 8−14
binding of default 8−12
conversion 8−9
declaration, default 8−11
evaluation of default 8−12/13
evaluation, order of 5−6
evaluation, unspecified order of 5−6
example of default 8−11/12
list, empty 8−9
list, variable 8−9
matching —see overload resolution
overloaded operator and default 13−20
passing 5−5
passing, reference and 8−21
reference 5−5
scope of default 8−13
specification, template 14−45
substitution 16−5
template 14−6
to constructor, unspecified 5−18
type checking 5−5
type checking of default 8−12
type, unknown 8−9
argument-dependent lookup 3−12
arguments, implementation-defined order of evaluation of function 8−13
`argv[]` 3−22
arithmetic
conversions, usual 5−2
exception 5−1
exception, undefined 5−1
pointer 5−22
type 3−33
`unsigned` 3−32
array
bound 8−8
`const` 3−34
declaration 8−8
declarator `[]` 8−8
declarator, multidimensional 8−8
delete 5−19
example 8−8
initialization 8−18
member 9−4
multidimensional 8−9
`new` 5−16
of class objects and constructor 12−12
of class objects and `new` 5−18
of class objects initialization 8−20, 12−12
order of execution, constructor and 12−12
order of execution, destructor and 12−8
overloading and pointer versus 13−2
parameter adjustment 8−10

pointer conversion 4−2
size, default 8−8
sizeof 5−15
storage of 8−9
type 3−33, 8−10
array-to-pointer conversion 4−2
arrow operator —see class member access operator
as-if rule 1−5
asin 26−20, 33
asm
declaration 7−23
implementation-defined 7−23
assembler 7−23
<assert.h> 17−9/D−1
assign
basic_string 21−18
deque 23−14
list 23−18
vector 23−26
Assignable requirements 23−1
assignment
and initialization, overloaded 12−12
and lvalue 5−27
conversion by 5−27
expression 5−27
operator 5−27, 17−7
operator, copy 12−21
operator, overloaded 13−21
operator restriction, copy 12−22
reference 8−21
to class object 5−27
to reference 5−27
assignment-expression 5−27
assignment-operator 5−27
at, basic_string 21−16
atan 26−20, 33
atan2 26−20, 33
atexit 3−24, 17−9, 18−9
auto 7−3
destruction of 6−5/6
initialization 6−7
object initialization 8−15
restriction 7−3
specifier 7−3
storage duration 3−25
automatic initialization 6−6/7
auto_ptr 20−18
auto_ptr 20−19
auto_ptr 20−19
get 20−20
operator* 20−20
operator-> 20−20
operator= 20−19
release 20−20
~auto_ptr 20−19
~auto_ptr, auto_ptr 20−19

B

back_inserter 24−16
back_insert_iterator 24−16
back_insert_iterator 24−16
back_insert_iterator 24−16
operator* 24−16
operator++ 24−16
operator= 24−16
backslash character 2−9
backspace 2−9
bad, basic_ios 27−20
bad_alloc 5−17, 18−11, 14
bad_alloc 18−14
bad_alloc 18−14
operator= 18−14
what 18−14
bad_alloc::what, implementation-defined 18−14
bad_cast 5−9, 18−16
bad_cast 18−16
bad_cast 18−16
operator= 18−16
what 18−16
bad_cast::what, implementation-defined 18−16
bad_exception 18−18
bad_exception 18−18
bad_exception 18−18
operator= 18−18
what 18−18
bad_exception::what, implementation-defined 18−18
bad_typeid 5−9, 18−16
bad_typeid 18−16
bad_typeid 18−16
operator= 18−16
what 18−16
bad_typeid::what, implementation-defined 18−17
base
class 17−11, 14
class 10−1/2
class access 11−3
class cast 5−11
class constructor order of execution 12−2
class destructor order of execution 12−8
class, direct 10−1
class, indirect 10−1
class initialization 12−13/14
class initialization, order of 12−14
class initializer 8−15
class member access 10−1
class member access, adjusting 11−4
class member ambiguity 10−4
class, private 11−3
class, public 11−3
class virtual —see virtual base class
of integer literal 2−8
base-specifier 10−1
base-specifier-list 10−1
basic
execution character set 1−4
source character set 2−2
basic_filebuf 27−4, 59
basic_filebuf 27−60
basic_filebuf 27−60
close 27−61, 68
imbue 27−64
is_open 27−61, 68
open 27−61, 68
overflow 27−62
pbackfail 27−62
rdbuf 27−68
seekoff 27−63
seekpos 27−64
setbuf 27−63
showmanyc 27−62
sync 27−64
uflow 27−62
underflow 27−62
~basic_filebuf 27−60
~basic_filebuf, basic_filebuf 27−60
basic_filebuf<char> 27−59
basic_filebuf<wchar_t> 27−59
basic_fstream 27−4, 67
basic_fstream 27−67
basic_fstream 27−67

basic_ifstream 27−4, 64
 basic_ifstream 27−65
 basic_ifstream 27−65
 close 27−65
 is_open 27−65
 open 27−65
 rdbuf 27−65
basic_ifstream<char> 27−59
basic_ifstream<wchar_t> 27−59
basic_ios 27−4, 16
 bad 27−20
 basic_ios 27−17
 basic_ios 27−17
 clear 27−19
 copyfmt 27−19
 eof 27−19
 exceptions 27−20
 fail 27−20
 fill 27−18
 good 27−19
 imbue 27−18
 init 27−34, 44
 narrow 27−18
 operator bool 27−19
 operator! 27−19
 rdbuf 27−18
 rdstate 27−19
 setstate 27−19
 tie 27−18
 widen 27−18
basic_ios<char> 27−7
basic_ios::failure argument, implementation-defined
   27−19
basic_iostream 27−41
 basic_iostream 27−41
 basic_iostream 27−41
 ~basic_iostream 27−41
~basic_iostream, basic_iostream 27−41
basic_ios<wchar_t> 27−7
basic_istream 27−4, 31
 basic_istream 27−33
 basic_istream 27−33
 gcount 27−37
 get 27−37
 getline 27−38/39
 ignore 27−39
 operator bool() 27−35
 operator>> 27−36
 peek 27−40
 putback 27−40
 read 27−40
 readsome 27−40
 seekg 27−40
 sentry 27−34
 sync 27−40
 tellg 27−40
 unget 27−40
 ~sentry 27−35
basic_istreambuf_iterator 27−4
basic_istream<char> 27−31
basic_istream<wchar_t> 27−31
basic_istringstream 27−4, 55
 basic_istringstream 27−55
 basic_istringstream 27−55
 rdbuf 27−56
 str 27−56
basic_istringstream<char> 27−51
basic_istringstream<wchar_t> 27−51
basic_ofstream 27−4, 66
 basic_ofstream 27−66
 basic_ofstream 27−66
 close 27−67
 is_open 27−66
 open 27−67
 rdbuf 27−66
basic_ofstream<char> 27−59
basic_ofstream<wchar_t> 27−59
basic_ostream 27−4
 basic_ostream 27−43
 basic_ostream 27−43
 flush 27−48
 operator bool() 27−44
 operator<< 27−45/46
 put 27−47
 seekp 27−44
 sentry 27−44
 tellp 27−44
 write 27−47
 ~basic_ostream 27−44
 ~sentry 27−44
~basic_ostream, basic_ostream 27−44
basic_ostreambuf_iterator 27−4
basic_ostream<char> 27−31
basic_ostream<wchar_t> 27−31
basic_ostringstream 27−4, 56
 basic_ostringstream 27−57
 basic_ostringstream 27−57
 rdbuf 27−57
 str 27−57
basic_ostringstream<char> 27−51
basic_ostringstream<wchar_t> 27−51
basic_streambuf 27−4, 23
 basic_streambuf 27−25
 basic_streambuf 27−25
 eback 27−27
 egptr 27−27
 epptr 27−27
 gbump 27−27
 getloc 27−25
 gptr 27−27
 imbue 27−27
 in_avail 27−26
 overflow 27−30
 pbackfail 27−29
 pbase 27−27
 pbump 27−27
 pptr 27−27
 pubimbue 27−25
 pubseekoff 27−25
 pubseekpos 27−26
 pubsetbuf 27−25
 pubsync 27−26
 sbumpc 27−26
 seekoff 27−28
 seekpos 27−28
 setbuf 27−28
 setg 27−27
 setp 27−27
 sgetc 27−26
 sgetn 27−26
 showmanyc 27−28, 62
 snextc 27−26
 sputbackc 27−26
 sputc 27−26
 sputn 27−26
 sungetc 27−26
 sync 27−28
 uflow 27−29
 underflow 27−29
 xsgetn 27−28
 xsputn 27−30
basic_streambuf<char> 27−22

basic_streambuf<wchar_t> 27−22
basic_string 21−9, 26, 27−50
 append 21−17
 assign 21−18
 at 21−16
 basic_string 21−12
 basic_string 21−12
 begin 21−15
 c_str 21−21
 capacity 21−16
 clear 21−16
 compare 21−25
 copy 21−21
 data 21−21
 empty 21−16
 end 21−15
 erase 21−19
 find 21−22
 find_first_not_of 21−24
 find_first_of 21−23
 find_last_not_of 21−24
 find_last_of 21−23
 getline 21−28
 insert 21−18
 max_size 21−15
 operator!= 21−27
 operator+ 21−26
 operator+= 21−17
 operator< 21−27
 operator<< 21−28
 operator<= 21−28
 operator= 21−14
 operator== 21−26
 operator> 21−27
 operator>= 21−28
 operator>> 21−28
 operator[] 21−16
 rbegin 21−15
 rend 21−15
 replace 21−20
 reserve 21−16
 resize 21−16
 rfind 21−22
 size 21−15
 substr 21−25
 swap 21−21, 28
basic_stringbuf 27−4, 51
 basic_stringbuf 27−52
 basic_stringbuf 27−52
 overflow 27−53
 pbackfail 27−53
 seekoff 27−53
 seekpos 27−54
 str 27−52
 underflow 27−53
basic_stringbuf<char> 27−51
basic_stringbuf<wchar_t> 27−51
basic_stringstream 27−4, 57
 basic_stringstream 27−58
 basic_stringstream 27−58
 rdbuf 27−58
 str 27−58
before, type_info 18−15
begin, basic_string 21−15
behavior
 default 17−2, 5
 implementation-defined 1−3, 22−37
 locale-specific 1−3
 reentrancy, implementation-defined 17−14
 required 17−2, 5
 undefined 1−3
 unspecified 1−3
Ben 13−3
bidirectional_iterator_tag 24−9/10
binary
 operator, interpretation of 13−21
 operator, overloaded 13−21
binary_function 20−9
binary_negate 20−11
binary_search 25−21
bind1st 20−12
bind2nd 20−12
binder1st 20−11
binder2nd 20−12
binding
 —see virtual function, dynamic
 of default argument 8−12
 reference 8−21
bit-field 9−11
 address of 9−11
 alignment of 9−11
 allocation, implementation defined 9−11
 declaration 9−11
 implementation defined alignment of 9−11
 implementation-defined 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 9−11
bitmask type 17−5/6
<bitset> 23−43
bitset 23−43
 any 23−47
 bitset 23−45
 bitset 23−45
 count 23−47
 flip 23−46
 none 23−47
 operator!= 23−47
 operator& 23−47
 operator&= 23−45
 operator<< 23−47/48
 operator<<= 23−45
 operator== 23−47
 operator>> 23−47/48
 operator>>= 23−46
 operator^ 23−48
 operator^= 23−45
 operator| 23−47
 operator|= 23−45
 operator~ 23−46
 reset 23−46
 set 23−46
 size 23−47
 test 23−47
 to_string 23−47
 to_ulong 23−46
bitwise
 AND operator 5−25
 exclusive OR operator 5−25
 inclusive OR operator 5−25
 operator 5−25
block
 initialization in 6−6
 scope —see local scope
 statement {} 6−1
 structure 6−6
body, function 8−14

bool promotion to int 4−3
bool
  increment 5−8, 15
  type-specifier 7−8
bool()
  basic_istream operator 27−35
  basic_ostream operator 27−44
boolalpha 27−20
Boolean
  bit-fields 9−11
  conversion 4−5
  literal 2−12
  type 3−32
  type 3−32
boolean-literal 2−12
bound array 8−8
bound, of array 8−8
break statement 6−5/6
built-in type —see fundamental type
byte 5−15
  string, null-terminated 17−6

C

C
  header 17−10/11, 17−13/D−1
  library, Standard 17−1, 6, 8, 17−10/C−10, C−12
  linkage to 7−24
  summary, compatibility with ISO C−1
call
  —see also function call, member function call, overloaded function call, virtual function call
  by reference 5−5
  by value 5−5
  operator function 13−20
  pseudo destructor 5−6
calloc 20−20/C−13
candidate functions 14−33
capacity
  basic_string 21−16
  vector 23−26
carriage return 2−9
case label 6−1, 3
<cassert> 17−9, 19−4
cast
  ambiguity, declaration versus 8−3
  base class 5−11
  const 5−12
  derived class 5−11
  dynamic 5−8, 18−16
  integer to pointer 5−11
  lvalue 5−10/11
  operator 5−13, 20, 8−2
  pointer to function 5−11
  pointer to integer 5−11
  pointer to member 5−11/12
  reference 5−10, 12
  reinterpret 5−11
  reinterpret_cast, lvalue 5−11
  reinterpret_cast, reference 5−12
  static 5−10
  static_cast, lvalue 5−10
  static_cast, reference 5−10
  to incomplete class 5−20
  undefined pointer to function 5−11
cast-expression 5−20
casting 5−6, 20
catch 15−1
category, locale 22−4
c-char 2−8
c-char-sequence 2−8
<cctype> 21−30
ceil 26−33
cerr 27−5
<cerrno> 17−11, 19−4
<cfloat> 18−9
C++
  Standard Library 17−1, 11, 13/14
  Standard Library exception specifications 17−15
  Standard library 17−11
  headers 17−8
change
  to const object, undefined 7−7
  to string literal, undefined 2−11
char
  implementation-defined sign of 3−32
  literal, implementation-defined value of 2−10
  type 3−32
  type, signed 3−32
  type specifier 7−8
  type, unsigned 3−32
character
  array initialization 8−20
  decimal-point 17−6
  literal 2−9
  literal, type of 2−9
  multibyte 1−3
  set, basic execution 1−4
  set, basic source 2−2
  signed 3−32
  string 2−11
  type 3−32
  underscore 17−11
character-literal 2−8
char_traits
  eq 21−22/24
  length 21−14/15, 17/20, 22/24, 26/27
checking
  point of error 14−25
  syntax 14−25
cin 27−5
<ciso646> C−12
class 3−33, 9−1
  abstract 10−10
  access and friend 11−6
  and type 9−1
  base 17−11, 14
  base —see base class
  cast to incomplete 5−20
  constructor and abstract 10−11
  conversion 12−4
  conversion ambiguity 10−6
  declaration, forward 9−2
  declaration {} 9−1
  definition 9−1, 4
  definition 3−2
  definition, empty 9−1
  definition example 9−4
  definition name hiding 9−2
  definition, scope of 9−2
  definition {} 9−1
  derived 17−14
  derived —see derived class
  gslice 26−23
  linkage of 3−20
  linkage specification 7−24
  local —see local class
  member —see also member
  member access 5−6
  member access operator 5−6
  member declaration 9−3

member function 9−5
member initialization 8−16
member semantics 5−6
member, `static` 3−25
member storage duration 3−27
member syntax 5−6
name 8−2
name as type definition 9−2
name declaration 3−1
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 5−27
object, `const` 3−34
object copy 12−19
object copy —see also copy constructor
object initialization 8−18, 12−12
object initialization —see also constructor
object layout 9−5, 10−2
object, member 9−4
object, operations on 9−1
object, `sizeof` 5−15
objects and constructor, array of 12−12
objects and `new`, array of 5−18
objects initialization, array of 8−20, 12−12
pointer to abstract 10−10
polymorphic 10−6
scope 3−7
scope of enumerator 7−11
`sizeof`, empty 9−1
template 23−44
template partial specializations 14−16
template specialization 14−5
unnamed 7−6
`class`
type specifier 9−1
versus `struct` 9−1
versus `union` 9−1
`classic`, `locale` 22−9
`classic_table`, `ctype<char>` 22−16
class-key 9−1
class-name 9−1
class-specifier 9−1
`clear`
`basic_ios` 27−19
`basic_string` 21−16
`<climits>` 18−9, 23−45/D−6
`<clocale>` 17−6, 22−48/C−13
`clog` 27−5
`close`
`basic_filebuf` 27−61, 68
`basic_ifstream` 27−65
`basic_ofstream` 27−67
`messages` 22−44
`<cmath>` 26−31
`codecvt` 22−17
`always_noconv` 22−18
`do_always_noconv` 22−20
`do_encoding` 22−20
`do_in` 22−19
`do_length` 22−20
`do_max_length` 22−20
`do_out` 22−19
`do_unshift` 22−19
`encoding` 22−18
`in` 22−18
`length` 22−18
`max_length` 22−19
`out` 22−18
`unshift` 22−18
`codecvt_byname` 22−20
`collate` 22−31
`compare` 22−32
`do_compare` 22−32
`do_hash` 22−32
`do_transform` 22−32
`hash` 22−32
`transform` 22−32
`collate_byname` 22−33
`combine`, `locale` 22−8
comma
operator 5−28
operator, side effects and 5−28
comment 2−3
`/* */` 2−5
`//` 2−5
`compare`
`basic_string` 21−25
`collate` 22−32
comparison
function 17−1
pointer 5−24
pointer to function 5−24
undefined pointer 5−22, 24
unspecified pointer 5−24
`void*` pointer 5−24
compatibility with ISO C summary C−1
compilation, separate 2−1
compiler control line —see preprocessing directive
complete object 1−4
completely defined object type 9−4
`<complex>` 26−2
`complex` 26−3
`abs` 26−8
`arg` 26−8
`complex` 26−5
`complex` 26−5
`conj` 26−8
`cos` 26−8
`cosh` 26−8
`exp` 26−8
`imag` 26−8
`log` 26−8
`log10` 26−9
`norm` 26−8
`operator!=` 26−7
`operator*` 26−7
`operator*=` 26−6
`operator+` 26−6
`operator+=` 26−6
`operator-` 26−6
`operator-=` 26−6
`operator/=` 26−6
`operator<<` 26−7
`operator==` 26−7
`operator>>` 26−7
`polar` 26−8
`pow` 26−9
`real` 26−8
`sin` 26−9
`sinh` 26−9
`sqrt` 26−9
`tan` 26−9
`tanh` 26−9
component 17−1
compound
statement 6−1
type 3−33
compound-statement 6−1
concatenation

string 2−11
undefined string literal 2−11
condition 6−2
conditional
expression operator 5−26
inclusion 16−2
conditional-expression, throw-expression in 5−26
*condition*s, rules for 6−2
`conj`, `complex` 26−8
consistency
example, linkage 7−3
linkage 7−3
linkage specification 7−25
type declaration 3−22
const
cast 5−12
member initialization 12−14
`*const` example 8−5
`const` 3−34
array 3−34
class object 3−34
constructor and 9−8, 12−1
destructor and 9−8, 12−7
example 8−5
initialization 7−7, 8−18
linkage of 3−20, 7−3
member function 9−7/8
object, undefined change to 7−7
overloading and 13−2
reference 8−22
type 7−6
constant 2−7, 5−2
enumeration 7−10
expression 5−28
expression, pointer to member 5−14
initializer 9−4
null pointer 4−4/5
pointer declaration 8−5
pointer example 8−5
constant-expression 5−28
constant-initializer 9−4
constructor 12−1
address of 12−2
and abstract class 10−11
and array order of execution 12−12
and `const` 9−8, 12−1
and initialization 12−12
and initialization example 12−12
and member function 12−2
and `new` 5−18
and `new`, unspecified 5−18
and `return` 6−6
and `static` objects order of execution 12−13
and virtual function call 12−17
and `volatile` 9−8, 12−1
array of class objects and 12−12
call, explicit 12−2
conversion by 12−5
conversion by —see also user-defined conversion
copy 12−2/3, 19, 17−7
default —see default constructor
definition 8−15
example 12−2
exception handling 15−4
for temporary 12−3
inheritance of 12−2
non-trivial 12−2
order of execution, base class 12−2
order of execution, member 12−2
restriction 12−1/2
restriction, copy 12−20
type of 12−2
`union` 9−10
unspecified argument to 5−18
container requirements 23−1
context, nondeduced 14−49
`continue`
in `for` statement 6−5
statement 6−5/6
control line —see preprocessing directive
convention 17−5
conversion
Boolean 4−5
ambiguity, class 10−6
and name hiding, user-defined 12−6
argument 8−9
array pointer 4−2
array-to-pointer 4−2
by assignment 5−27
by constructor 12−5
class 12−4
derived-to-base 13−14
explicit type —see casting
floating point 4−4
floating-integral 4−4
function 12−6
function —see also user-defined conversion
function-to-pointer 4−2
implementation defined pointer integer 5−11
implementation-defined floating point 4−4
implicit 4−1, 12−4
implicit user-defined 12−5
inheritance of user-defined 12−7
integer 4−4
lvalue-to-rvalue 4−2
operator —see conversion function
overload resolution and 13−12
overload resolution and pointer 13−19
pointer 4−4
pointer to function 4−2
pointer to member 4−5
pointer to member `void*` 4−5
rank 13−14
return type 6−6
`reverse_iterator` 24−13
sequence, implicit 13−13
sequence, standard 4−1
`signed unsigned` integer 4−4
standard 4−1
to enumeration type 5−11
to enumeration type, `static_cast`, 5−11
to rvalue, lvalue 4−2
type of 12−6
undefined floating point 4−4
user-defined 12−4/6
virtual user-defined 12−7
conversion-function-id 12−6
conversions
qualification 4−2
usual arithmetic 5−2
copy
assignment operator 12−21
assignment operator 12−19
assignment operator, implicitly-declared 12−21
assignment operator restriction 12−22
class object 12−19
constructor 12−2/3, 19, 17−7
constructor, implicitly-declared 12−20
constructor restriction 12−20
initialization 8−17
`copy` 25−13
`basic_string` 21−21

copy_backward 25−13
CopyConstructible requirements 20−2
copyfmt, basic_ios 27−19
cos 26−20, 33
 complex 26−8
cosh 26−20, 33
 complex 26−8
count 25−11
 bitset 23−47
count_if 25−11
cout 27−5
__cplusplus 16−9
<csetjmp> 17−11, 18−20
cshift, valarray 26−17
<csignal> 18−20
<cstdarg> 8−10, 17−11, 18−20
<cstddef> 5−15, 22, 18−1/C−13
<cstdio> 27−5/6, 59, 61, 63, 27−68/C−13
<cstdlib> 3−22, 24, 17−9, 18−9, 20, 20−20, 21−31, 25−27, 26−32/C−13
c_str, basic_string 21−21
<cstring> 17−6, 20−20, 21−30/D−6, D−11/C−13
<ctime> 18−20, 20−21, 22−2/C−13
ctor-initializer 12−13
ctype 22−10
 do_is 22−12
 do_narrow 22−13
 do_scan_is 22−12
 do_scan_not 22−12
 do_tolower 22−12
 do_toupper 22−12
 do_widen 22−13
 is 22−11
 narrow 22−12
 scan_is 22−11
 scan_not 22−11
 tolower 22−12
 toupper 22−11
 widen 22−12
ctype_byname 22−13
ctype_byname<char> 22−17
ctype<char>
 classic_table 22−16
 ctype<char> 22−15
 ctype<char> 22−15
 is 22−15
 narrow 22−16
 scan_is 22−15
 scan_not 22−16
 table 22−16
 tolower 22−16
 toupper 22−16
 widen 22−16
 ~ctype<char> 22−15
~ctype<char>, ctype<char> 22−15
<ctype.h> D−1
cv-qualified name, address of 5−14
cv-qualifier 3−34
cv-qualifier 8−2
<cwchar> 17−7, 11, 21−30/C−13
<cwctype> 17−11, 21−30

D

DAG
 multiple inheritance 10−3
 nonvirtual base class 10−3
 virtual base class 10−3
data
 member —see member
 member, static 9−8
data, basic_string 21−21
date_order, time_get 22−34
deallocation
 —see delete
 function 3−26, 5−19, 12−10
dec 27−21, 46
decimal literal 2−8
decimal-literal 2−7
decimal-point character 17−6
decimal_point, numpunct 22−30
declaration 3−1, 7−1
 :, field 9−11
 access 11−4
 ambiguity, function 8−16
 array 8−8
 as definition 7−2
 asm 7−23
 bit-field 9−11
 class member 9−3
 class name 3−1
 class name, point of 9−3
 consistency, type 3−22
 constant pointer 8−5
 default argument 8−11
 definition versus 3−1
 ellipsis in function 5−5, 8−9
 enumerator point of 3−5
 example 3−2, 8−11
 example, function 8−10
 extern 3−1
 extern reference 8−21
 forward 7−4
 forward class 9−2
 function 3−1, 8−9
 hiding —see name hiding
 in for, scope of 6−5
 in for statement 6−5
 in switch statement 6−3
 matching, overloaded function 13−3
 member 9−3
 multiple 3−22
 name 3−1
 name, point of 3−5
 overloaded 13−1
 overloaded name and friend 11−6
 parameter 8−9
 parentheses in 8−3, 5
 pointer 8−5
 reference 8−6
 register 7−3
 specifier 7−2
 statement 6−6
 static member 3−1
 storage class 7−3
 type 8−4
 type ambiguity 7−2
 typedef 3−1
 typedef as type 7−5
 versus cast ambiguity 8−3
 versus expression ambiguity 6−7
 {}, class 9−1
 {}, enum 7−10
declaration 7−1
declaration-statement 6−6
declarative region 3−1, 4
declarator 7−1, 8−1
 &, reference 8−6
 (), function 8−9
 *, pointer 8−5
 ::*, pointer to member 8−7

[], array 8−8
example 8−2
initializer, temporary and 12−4
meaning of 8−4
multidimensional array 8−8
declarator 8−1
declarator-id 8−2
decl-specifier 7−2
decrement
operator 5−8, 14/15
operator, overloaded 13−22
default
access control 11−1
argument and name hiding 8−13
argument and virtual function 8−14
argument, binding of 8−12
argument declaration 8−11
argument, evaluation of 8−12/13
argument, example of 8−11/12
argument, overload resolution and 13−11
argument, overloaded operator and 13−20
argument, scope of 8−13
argument, type checking of 8−12
array size 8−8
behavior 17−2, 5
constructor 12−2
constructor and initialization 12−12
constructor and `new` 5−18
destructor 12−7
initialization 8−16
initializers, overloading and 13−3
member access, `struct` 9−1
member access, `union` 9−1
`default` label 6−1, 3
default-initialization 8−16
`#define` 16−5
definition 3−1, 17−1
altermate 17−11
and initialization 7−2
class 3−2
class 9−1, 4
class name as type 9−2
constructor 8−15
declaration as 7−2
empty class 9−1
enumerator 3−2
enumerator point of 7−10
example 3−1
example, function 8−15
example, nested class 9−12, 11−10
function 3−2
function 8−14
local class 9−13
member function 9−5
name hiding, class 9−2
namespace 7−12
nested class 9−12
object 3−2
of template 14−1
pure virtual function 10−10
scope, macro 16−6
scope of class 9−2
`static` member 9−9
versus declaration 3−1
virtual function 10−8
{}, class 9−1
definitions, implementation-generated 3−2
delete
array 5−19
object 5−19
`delete` 3−25, 5−19, 12−10
destructor and 5−19, 12−8
example 12−10
example, destructor and 12−11
example, scope of 12−11
`operator` 17−12, 18−12, 20−20
overloading and 3−27
type of 12−10
undefined 5−19
`delete[]`, `operator` 17−12, 18−13
deleted object, undefined 3−27
delete-expression 5−19
dependent name 14−29, 32
deprecated features 5−8, 15
`<deque>` 23−9
`deque` 23−12
`assign` 23−14
`erase` 23−15
`insert` 23−15
`resize` 23−15
dereferencing —see also indirection
derivation —see inheritance
derived
class 17−14
class 10−1
class cast 5−11
class example 10−1
class, most 1−5
class, overloading and 13−3
object, most 1−5
derived-to-base conversion 13−14
destination type 8−17
destruction
of `auto` 6−5/6
of local `static` 6−7
of local variable 6−5/6
of temporary 12−3
of temporary, order of 12−4
destructor 12−7, 17−7
and array order of execution 12−8
and `const` 9−8, 12−7
and `delete` 5−19, 12−8
and `delete` example 12−11
and exception, explicit 12−9
and exit from scope 6−5
and fundamental type 12−9
and member function 12−8
and placement of object 12−9
and virtual function call 12−17
and `volatile` 9−8, 12−7
call example, explicit 12−9
call, explicit 12−8
call, implicit 12−8
call, pseudo 5−6
default 12−7
exception handling 15−4
for temporary 12−3
non-trivial 12−7
order of execution 12−8
order of execution, base class 12−8
order of execution, member 12−8
program termination and 12−8
pure virtual 12−8
restriction 12−7/8
`static` object 3−24
`union` 9−10
virtual 12−8
diagnostic message 1−2
digit 2−6
digit-sequence 2−10
digraph 2−4
direct

base class 10−1
binding of reference 8−22
initialization 8−17
direct-abstract-declarator 8−2
direct-declarator 8−1
directed acyclic graph —see DAG
directive
error 16−8
null 16−9
pragma 16−9
preprocessing 16−1
direct-new-declarator 5−16
`distance` 24−11
distinct string 2−11
`div` 26−32
`divides` 20−9
division
by zero, undefined 5−1, 22
implementation defined 5−22
operator 5−21
`djacent_difference` 26−31
`do` statement 6−3, 5
`do_always_noconv`, `codecvt` 22−20
`do_close`, `messages` 22−45
`do_compare`, `collate` 22−32
`do_curr_symbol`, `moneypunct` 22−42
`do_date_order`, `time_get` 22−35
`do_decimal_point`
`moneypunct` 22−42
`numpunct` 22−31
`do_encoding`, `codecvt` 22−20
`do_falsename`, `numpunct` `do_truename` 22−31
`do_frac_digits`, `moneypunct` 22−43
`do_get`
`messages` 22−44
`money_get` 22−38
`num_get` 22−23
`do_get_date`, `time_get` 22−35
`do_get_monthname`, `time_get` 22−35
`do_get_time`, `time_get` 22−35
`do_get_weekday`, `time_get` 22−35
`do_get_year`, `time_get` 22−35
`do_grouping`
`moneypunct` 22−42
`numpunct` 22−31
`do_hash`, `collate` 22−32
`do_in`, `codecvt` 22−19
`do_is`, `ctype` 22−12
`do_length`, `codecvt` 22−20
`domain_error` 19−2
`domain_error` 19−2
`domain_error` 19−2
`do_max_length`, `codecvt` 22−20
dominance, virtual base class 10−5
`donarrow` 22−16
`do_narrow`, `ctype` 22−13
`do_negative_sign`, `moneypunct` 22−43
`do_neg_format`, `moneypunct` 22−43
`do_open`, `messages` 22−44
`do_out`, `codecvt` 22−19
`do_pos_format`, `moneypunct` 22−43
`do_positive_sign`, `moneypunct` 22−43
`do_put`
`money_put` 22−40
`num_put` 22−26
`time_put` 22−37
`do_scan_is`, `ctype` 22−12
`do_scan_not`, `ctype` 22−12
dot operator —see class member access operator
`do_thousands_sep`
`moneypunct` 22−42
`numpunct` 22−31
`do_tolower`, `ctype` 22−12
`do_toupper`, `ctype` 22−12
`do_transform`, `collate` 22−32
`do_truename` `do_falsename`, `numpunct` 22−31
double quote 2−9
`double`
literal 2−11
type 3−32
type specifier 7−8
`do_unshift`, `codecvt` 22−19
`dowiden` 22−16
`do_widen`, `ctype` 22−13
dynamic
binding —see virtual function
cast 5−8, 18−16
initialization 3−23
storage duration 3−25, 5−16
type 1−2

E

`E` suffix 2−11
`eback`, `basic_streambuf` 27−27
effect, side 1−5
`egptr`, `basic_streambuf` 27−27
elaborated
class name 7−9, 9−2/3
`enum` name 7−9
type specifier —see elaborated class name
elaborated-type-specifier 7−9
`#elif` 16−2
elimination of temporary 12−3
ellipsis
example 8−9
in function declaration 5−5, 8−9
overload resolution and 13−11
`#else` 16−3
`else` 6−2
empty
argument list 8−9
class definition 9−1
class `sizeof` 9−1
statement 6−1
`empty` 24−10
`basic_string` 21−16
encoding, multibyte 2−12
`encoding`, `codecvt` 22−18
`end`, `basic_string` 21−15
`#endif` 16−3
`endl` 27−46, 48
end-of-file 23−48
`ends` 27−48
entity 3−1
enum name, `typedef` 7−6
`enum` 3−33
declaration `{}` 7−10
name, elaborated 7−9
overloading and 13−2
type of 7−10
type specifier 7−9
enumerated type 3−33, 17−5
enumeration 7−10
constant 7−10
example 7−11
linkage of 3−20
type, conversion to 5−11
type, `static_cast`, conversion to 5−11
underlying type 7−11
enumerator

class, scope of 7−11
definition 3−2
member 7−11
point of declaration 3−5
point of definition 7−10
redefinition 7−10
restriction 7−10
value of 7−10
enumerator 7−10
environment, program 3−22
`eof`, `basic_ios` 27−19
`epptr`, `basic_streambuf` 27−27
`eq`, `char_traits` 21−22/24
`equal` 25−11
`istreambuf_iterator` 24−24
equality operator 5−24
`EqualityComparable` requirements 20−1
equality-expression 5−24
`equal_range` 25−21
`equal_to` 20−9
equivalence
template type 14−10
type 7−5, 9−2
equivalent
parameter declarations 13−2
parameter declarations, overloading and 13−2
`erase`
`basic_string` 21−19
`deque` 23−15
`list` 23−19
`vector` 23−27
`<errno.h>` D−1
error
checking, point of 14−25
directive 16−8
`#error` 16−8
escape
character —see backslash
sequence 2−9
sequence, undefined 2−10
escape-sequence 2−9
evaluation
`new`, unspecified order of 5−18
of default argument 8−12/13
of expression, order of 1−6
order of argument 5−6
unspecified order of 3−23, 5−1
unspecified order of argument 5−6
unspecified order of function call 5−6
example
`*const` 8−5
array 8−8
class definition 9−4
`const` 8−5
constant pointer 8−5
constructor 12−2
constructor and initialization 12−12
declaration 3−2, 8−11
declarator 8−2
definition 3−1
`delete` 12−10
derived class 10−1
destructor and `delete` 12−11
ellipsis 8−9
enumeration 7−11
explicit destructor call 12−9
explicit qualification 10−5
`friend` 9−3
`friend` function 11−5
function declaration 8−10
function definition 8−15
linkage consistency 7−3
local class 9−13
member function 9−7, 11−5
member name access 11−5
nested class 9−12
nested class definition 9−12, 11−10
nested class forward declaration 9−12
nested type name 9−13
of default argument 8−11/12
of incomplete type 3−31
of overloading 13−1
pointer to member 8−7
pure virtual function 10−10
scope of `delete` 12−11
scope resolution operator 10−5
`static` member 9−9
subscripting 8−8
type name 8−2
`typedef` 7−5
unnamed parameter 8−15
variable parameter list 8−9
virtual function 10−8
exception
allowing an 15−7
and `new` 5−18
arithmetic 5−1
declaration scope 3−6
explicit destructor and 12−9
handler 15−4, 17−14
handler, incomplete type in 15−4
handling 15−1
handling constructor 15−4
handling destructor 15−4
specifications, C++ Standard Library 17−15
specifications, Standard C library 17−15
specifications, implementation-defined 17−15
throwing 15−2
types, implementation-defined 17−15
undefined arithmetic 5−1
`<exception>` 18−17
`exception`
`exception` 18−17
`exception` 18−17
`operator=` 18−17
`what` 18−17
`~exception` 18−17
`~exception`, `exception` 18−17
exception-declaration 15−1
`exceptions`, `basic_ios` 27−20
exception-specification 15−6
`exception::what` message, implementation-defined 18−18
execution character set, basic 1−4
exit from scope, destructor and 6−5
`exit` 3−22, 24, 6−5, 17−9, 18−10, 14
`exp` 26−20, 33
`complex` 26−8
explanation, subscripting 8−8
explicit
constructor call 12−2
destructor and exception 12−9
destructor call 12−8
destructor call example 12−9
instantiation 14−38
qualification 3−13
qualification example 10−5
specialization, template 14−39
type conversion —see casting
`explicit` specifier 7−5
explicit-specialization 14−39
exponent-part 2−10

export 14−1
expression 5−1
 ambiguity, declaration versus 6−7
 assignment 5−27
 constant 5−28
 order of evaluation of 1−6
 parenthesized 5−3
 pointer to member constant 5−14
 postfix 5−4
 primary 5−2
 reference 5−1
 statement 6−1
 unary 5−13
expression 5−28
expression-list 5−4
expression-statement 6−1
`extern` 7−3
 `"C"` 17−10/11
 `"C++"` 17−10/11
 declaration 3−1
 linkage of 7−3
 linkage specification 7−23
 reference declaration 8−21
 restriction 7−3
external linkage 3−20, 17−10/11

F

`F` suffix 2−11
`f` suffix 2−11
`facet`, `locale` 22−6
`fail`, `basic_ios` 27−20
`failed`, `ostreambuf_iterator` 24−25
`failure`, `ios_base::failure` 27−10
`falsename`, `numpunct truename` 22−30
`fclose` 27−61
field declaration `:` 9−11
file 2−1
 source 2−1, 17−9, 11
`filebuf` 27−4, 59
 implementation-defined 27−64
`fill` 25−15
 `basic_ios` 27−18
 `gslice_array` 26−26
 `indirect_array` 26−29
 `mask_array` 26−27
 `slice_array` 26−23
`fill_n` 25−15
final overrider 10−7
`find` 25−10
 `basic_string` 21−22
`find_end` 25−10
`find_first_not_of`, `basic_string` 21−24
`find_first_of` 25−10
 `basic_string` 21−23
`find_if` 25−10
`find_last_not_of`, `basic_string` 21−24
`find_last_of`, `basic_string` 21−23
fIoctal-digit 2−8
`fixed` 27−22
`flags`, `ios_base` 22−10, 27−12
`flip`, `bitset` 23−46
`float`
 literal 2−11
 type 3−32
 type specifier 7−8
`<float.h>` D−1
floating
 point conversion 4−4
 point conversion, implementation-defined 4−4
 point conversion, undefined 4−4
 point literal 2−11
 point literal, type of 2−11
 point promotion 4−4
 point type 3−32
 point type 3−32
 point type, implementation-defined 3−33
floating-integral conversion 4−4
floating-literal 2−10
floating-suffix 2−11
`float_round_style` 18−7
`floor` 26−33
`flush` 27−12, 34, 44, 48
 `basic_ostream` 27−48
`fmtflags`
 `ios` 27−49
 `ios_base` 27−10
`fopen` 27−61
`for`
 scope of declaration in 6−5
 statement 6−3, 5
 statement, `continue` in 6−5
 statement, declaration in 6−5
`for_each` 25−9
form feed 2−9
formal argument —see parameter
forward
 class declaration 9−2
 declaration 7−4
 declaration example, nested class 9−12
`forward_iterator_tag` 24−9/10
`fpos` 27−7, 15
 `fpos` 27−15
 `fpos` 27−15
 `state` 27−15
fractional-constant 2−10
free store —see also `new`, `delete`
`free` 20−20
freestanding implementation 17−9
`freeze`
 `ostrstream` D−11
 `strstreambuf` D−6
`frexp` 26−33
friend
 local class 11−7
 specifier 17−14
`friend`
 access specifier and 11−7
 class access and 11−6
 declaration, overloaded name and 11−6
 example 9−3
 function, access and 11−5
 function example 11−5
 function, inline 11−7
 function, linkage of 11−6
 function, member function and 11−5
 function, nested class 9−13
 inheritance and 11−7
 local class and 11−7
 member function 11−6
 specifier 7−6
 template and 14−15
 `virtual` and 10−8
`front_inserter` 24−17
`front_insert_iterator` 24−17
 `front_insert_iterator` 24−17
 `front_insert_iterator` 24−17
 `operator*` 24−17
 `operator++` 24−17
 `operator=` 24−17
`fseek` 27−61

<fstream> 27−58
fstream 27−4
full-expression 1−6
function
 —see also friend function, member function, inline function, virtual function
 allocation 3−26, 5−17, 12−9
 argument —see argument
 arguments, implementation-defined order of evaluation of 8−13
 body 8−14
 call 5−5
 call evaluation, unspecified order of 5−6
 call operator 5−5, 13−20
 call operator, overloaded 13−21
 call, recursive 5−6
 call, undefined 5−11/12
 cast, pointer to 5−11
 cast, undefined pointer to 5−11
 comparison 17−1
 comparison, pointer to 5−24
 conversion 12−6
 conversion, pointer to 4−2
 deallocation 3−26, 5−19, 12−10
 declaration 3−1, 8−9
 declaration ambiguity 8−16
 declaration, ellipsis in 5−5, 8−9
 declaration example 8−10
 declaration matching, overloaded 13−3
 declarator () 8−9
 definition 8−14
 definition 3−2
 definition example 8−15
 global 17−11, 13/14
 handler 17−2
 linkage specification overloaded 7−25
 modifier 17−2
 name hiding 13−3
 name, overloaded 13−1
 observer 17−2
 operator 13−19
 overloaded —see also overloading
 parameter —see parameter
 parameter adjustment 8−10
 plain old 18−21
 pointer to member 5−21
 prototype scope 3−6
 replacement 17−2
 reserved 17−2
 return —see return
 return type —see return type
 scope 3−6
 specifier 7−4
 template 14−45
 template overload resolution 14−53
 template partial ordering 14−22
 type 3−33, 8−9/10
 typedef 8−10
 viable 13−5
 virtual —see virtual function
 virtual member 17−11, 14
<functional> 20−7
function-body 8−14
function-definition 8−14
function-like macro 16−4
functions, candidate 14−33
function-specifier 7−4
function-to-pointer conversion 4−2
function-try-block 15−1
fundamental
 type 3−32
 type conversion —see conversion, user-defined conversion
 type, destructor and 12−9

G

gbump, basic_streambuf 27−27
gcount, basic_istream 27−37
generate 25−15
generated destructor —see default destructor
generate_n 25−15
get
 auto_ptr 20−20
 basic_istream 27−37
 money_get 22−38
 num_get 22−22
get_date, time_get 22−34
getline
 basic_istream 27−38/39
 basic_string 21−28
getloc
 basic_streambuf 27−25
 ios_base 27−13
get_monthname, time_get 22−34
get_temporary_buffer 20−17
get_time, time_get 22−34
get_weekday, time_get 22−34
get_year, time_get 22−35
global
 anonymous union 9−10
 function 17−11, 13/14
 name 3−7
 namespace 17−11
 namespace scope 3−7
 scope 3−7
global, locale 22−9
good, basic_ios 27−19
goto
 initialization and 6−6
 statement 6−1, 5/6
gptr, basic_streambuf 27−27
grammar A−1
greater
 than operator 5−23
 than or equal to operator 5−23
greater 20−10
greater_equal 20−10
grouping, numpunct 22−30
gslice
 class 26−23
 gslice 26−24
 gslice 26−24
 size 26−24
 start 26−24
 stride 26−24
gslice_array 26−25
 fill 26−26
 gslice_array 26−25
 gslice_array 26−25
 operator%= 26−26
 operator&= 26−26
 operator*= 26−26
 operator+= 26−26
 operator-= 26−26
 operator/= 26−26
 operator<<= 26−26
 operator= 26−25
 operator>>= 26−26
 operator^= 26−26
 operator|= 26−26

H

handler
exception 15−4, 17−14
function 17−2
incomplete type in exception 15−4
handler 15−1
handler-seq 15−1
`hasfacet`, `locale` 22−9
`hash`, `collate` 22−32
header, C 17−10/11, 17−13/D−1
header-name 2−5
headers, C++ 17−8
hex number 2−10
`hex` 27−22
hexadecimal literal 2−8
hexadecimal-digit 2−8
hexadecimal-escape-sequence 2−9
hexadecimal-literal 2−7
hex-quad 2−2
hiding —see name hiding
horizontal tab 2−9
hosted implementation 17−9

I

id, qualified 5−3
`id`, `locale` 22−7
identifier 2−6, 5−3, 7−1
_, underscore in 2−6
identifier 2−6
identities and overloading, operator 13−20
id-expression 5−3
id-expression 5−2
`#if` 16−2, 17−13
`if` statement 6−2/3
`#ifdef` 16−3
`#ifndef` 16−3
`ifstream` 27−4, 59
`ignore`, `basic_istream` 27−39
ill-formed program 1−3
`imag` 26−7
`complex` 26−8
`imbue`
`basic_filebuf` 27−64
`basic_ios` 27−18
`basic_streambuf` 27−27
`ios_base` 27−13
implementation
defined alignment of bit-field 9−11
defined bit-field allocation 9−11
defined division 5−22
defined modulus 5−22
defined pointer integer conversion 5−11
defined pointer subtraction 5−22
defined right shift 5−23
defined type of `ptrdiff_t` 5−22
freestanding 17−9
hosted 17−9
limits 1−3
implementation-defined 4−4, 8−22, 17−9, 11, 18−2, 10, 14, 16/18, 27−13, 55, 27−63/C−13
`__STDC__` 16−9
alignment requirement 3−31
`asm` 7−23
`bad_alloc::what` 18−14
`bad_cast::what` 18−16
`bad_exception::what` 18−18
`bad_typeid::what` 18−17
`basic_ios::failure` argument 27−19
behavior 1−3, 22−37
behavior reentrancy 17−14
exception specifications 17−15
exception types 17−15
`exception::what` message 18−18
`filebuf` 27−64
floating point conversion 4−4
floating point type 3−33
generation of temporary 12−3
linkage of `main()` 3−22
linkage specification 7−24
object linkage 7−26
order of evaluation of function arguments 8−13
parameters to `main()` 3−22
sign of bit-field 9−11
sign of `char` 3−32
`sizeof` integral type 3−32
`sizeof` type 3−32
`streambuf` 27−2
`streamoff` 27−7/D−2
`streampos` D−2
string literal 2−11
type of integer literal 2−8
`type_info::name` 18−15
types 17−5
value of `char` literal 2−10
value of multicharacter literal 2−9
`volatile` 7−8
`wchar_t` 3−32
implementation-dependent 27−34, 44
implementation-generated definitions 3−2
implementation-specified `smanip` 27−48
implicit
conversion 4−1, 12−4
conversion sequence 13−13
conversion sequences implied object parameter 13−6
destructor call 12−8
instantiation, template 14−35
object argument 13−5
user-defined conversion 12−5
implicitly-declared
copy assignment operator 12−21
copy constructor 12−20
default constructor 12−2
default constructor —see default constructor
implied
object parameter 13−5
object parameter, implicit conversion sequences 13−6
`in`, `codecvt` 22−18
`in_avail`, `basic_streambuf` 27−26
`#include` 16−3, 17−9
`includes` 25−23
inclusion
conditional 16−2
source file 16−3
incomplete
class, cast to 5−20
type 3−2/3, 5, 31, 4−2, 5−4/9, 14/16, 19, 22, 27, 10−1
type, example of 3−31
type in exception handler 15−4
increment
`bool` 5−8, 15
operator 5−8, 14/15
operator, overloaded 13−22
indeterminate uninitialized variable 8−16
indirect base class 10−1
`indirect_array` 26−27
`fill` 26−29
`indirect_array` 26−28
`indirect_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
indirection 5–14
operator 5–14
inequality operator 5–24
inheritance 10–1
—see also multiple inheritance
and friend 11–7
of constructor 12–2
of overloaded operator 13–20
of user-defined conversion 12–7
Init, ios_base::Init 27–12
init, basic_ios 27–34, 44
~Init, ios_base::Init 27–12
init-declarator 8–1
init-declarator-list 8–1
initialization 8–15
and goto 6–6
and new 5–18
array 8–18
array of class objects 8–20, 12–12
auto 6–7
auto object 8–15
automatic 6–6/7
base class 12–13/14
character array 8–20
class member 8–16
class object 8–18, 12–12
class object —see also constructor
const 7–7, 8–18
const member 12–14
constructor and 12–12
copy 8–17
default 8–16
default constructor and 12–12
definition and 7–2
direct 8–17
dynamic 3–23
example, constructor and 12–12
in block 6–6
jump past 6–3, 6
local static 6–7
member 12–13
member object 12–14
order of 3–23, 10–2
order of base class 12–14
order of member 12–14
order of virtual base class 12–14
overloaded assignment and 12–12
parameter 5–5
reference 8–7, 21
reference member 12–14
run-time 3–23
static member 9–9
static object 3–23, 8–15/16
struct 8–18
union 8–20, 9–10
virtual base class 12–15, 21
initializer 8–15
base class 8–15
constant 9–4
list {} 8–18
member 8–15
scope of member 12–15
temporary and declarator 12–4
initializer 8–15
initializer-clause 8–15
initializer-list 8–15
inline 17–13
friend function 11–7
function 7–5
member function 9–5
inline
linkage of 3–20
specifier 7–5
inner_product 26–30
inplace_merge 25–22
input_iterator_tag 24–9/10
insert
basic_string 21–18
deque 23–15
list 23–19
vector 23–27
inserter 24–19
insert_iterator 24–18
insert_iterator 24–18
insert_iterator 24–18
operator* 24–18
operator++ 24–18
operator= 24–18
instantiation
explicit 14–38
point of 14–32
template implicit 14–35
unit 2–2
int, bool promotion to 4–3
int
type 3–32
type specifier 7–8
type, unsigned 3–32
integer
cast, pointer to 5–11
conversion 4–4
conversion, implementation defined pointer 5–11
conversion, signed unsigned 4–4
literal 2–8
literal, base of 2–8
literal, implementation-defined type of 2–8
literal, type of 2–8
to pointer cast 5–11
type 3–32
integer-literal 2–7
integer-suffix 2–8
integral
promotion 4–3
type 3–32
type 3–32
type, implementation-defined sizeof 3–32
value, undefined unrepresentable 4–4
internal linkage 3–20
internal 27–21
interpretation
of binary operator 13–21
of unary operator 13–20
invalid_argument 19–2, 23–44/45
invalid_argument 19–2
invalid_argument 19–2
invocation, macro 16–5
<iomanip> 27–31
<ios> 27–6
ios 27–4, 7
fmtflags 27–49
ios_base 27–7
flags 22–10, 27–12
fmtflags 27–10

getloc 27−13
imbue 27−13
ios_base 27−14
ios_base 27−14
iostate 27−11
iword 27−14
openmode 27−11
precision 22−10, 27−13
pword 27−14
register_callback 27−14
seekdir 27−11
setf 27−12
sync_with_stdio 27−13
unsetf 27−13
width 22−10, 27−13
xalloc 27−14
ios_base::failure 27−10
failure 27−10
what 27−10
ios_base::Init 27−12
Init 27−12
~Init 27−12
<iosfwd> 27−2
iostate, ios_base 27−11
<iostream> 27−5
is
ctype 22−11
ctype<char> 22−15
isalnum 22−9
isalpha 22−9
iscntrl 22−9
isdigit 22−9
isgraph 22−9
islower 22−9
ISO C summary, compatibility with C−1
<iso646.h> D−1/C−12
is_open
basic_filebuf 27−61, 68
basic_ifstream 27−65
basic_ofstream 27−66
isprint 22−9
ispunct 22−9
isspace 22−9
<istream> 27−31
istream 27−4, 31
operator>> 27−35
istreambuf_iterator 24−22
equal 24−24
istreambuf_iterator 24−23
istreambuf_iterator 24−23
operator!= 24−24
operator* 24−23
operator++ 24−24
operator== 24−24
proxy 24−23
istream_iterator 24−19
operator== 24−21
istringstream 27−4, 51
istrstream D−10
istrstream D−10
istrstream D−10
rdbuf D−10
str D−10
isupper 22−9
isxdigit 22−9
iteration statement 6−3
iteration-statement 6−3, 6
scope 6−4
iterator requirements 24−1
<iterator> 24−6
iter_swap 25−13
iword, ios_base 27−14

J

Jessie 12−5
jump
past initialization 6−3, 6
statement 6−5
jump-statement 6−5

K

keyword A−1
list 2−6

L

L
prefix 2−9, 11
suffix 2−8, 11
l suffix 2−8, 11
label 6−6
case 6−1, 3
default 6−1, 3
name space 6−1
scope of 3−6, 6−1
specifier : 6−1
labeled statement 6−1
lattice —see DAG, sub-object
layout
access specifier and object 11−3
bit-field 9−11
class object 9−5, 10−2
layout-compatible type 3−31
left
shift operator 5−23
shift, undefined 5−23
left 27−21
length of name 2−6
length
char_traits 21−14/15, 17/20, 22/24, 26/27
codecvt 22−18
valarray 26−17
length_error 19−2, 21−9
length_error 19−3
length_error 19−3
less
than operator 5−23
than or equal to operator 5−23
less 20−10
less_equal 20−10
LessThanComparable requirements 20−1
lexical conventions 2−1
lexicographical_compare 25−26
Library, C++ Standard 17−1, 11, 13/14
library
C++ Standard 17−11
Standard C 17−1, 6, 8, 17−10/C−10, C−12
limits, implementation 1−3
<limits> 18−2
<limits.h> D−1
#line 16−8
linkage 3−1, 20
consistency 7−3
consistency example 7−3
external 3−20, 17−10/11
implementation-defined object 7−26
internal 3−20
of class 3−20

of `const` 3−20, 7−3
of enumeration 3−20
of `extern` 7−3
of `friend` function 11−6
of `inline` 3−20
of `main()`, implementation-defined 3−22
of `static` 3−20, 7−3
specification 7−23
specification class 7−24
specification consistency 7−25
specification, `extern` 7−23
specification, implementation-defined 7−24
specification object 7−26
specification overloaded function 7−25
to C 7−24
linkage-specification 7−23
list
keyword 2−6
operator 2−7, 13−19
`{}`, initializer 8−18
`<list>` 23−9
`list` 23−16
`assign` 23−18
`erase` 23−19
`insert` 23−19
`merge` 23−20
`remove` 23−20
`resize` 23−18
`reverse` 23−20
`sort` 23−20
`splice` 23−19
`unique` 23−20
literal 2−7, 5−2
base of integer 2−8
character 2−9
decimal 2−8
`double` 2−11
`float` 2−11
floating point 2−11
hexadecimal 2−8
implementation-defined type of integer 2−8
implementation-defined value of `char` 2−10
implementation-defined value of multicharacter 2−9
integer 2−8
`long` 2−8
`long double` 2−11
multicharacter 2−9
narrow-character 2−9
octal 2−8
type of character 2−9
type of floating point 2−11
type of integer 2−8
`unsigned` 2−8
literal 2−7
local
class and `friend` 11−7
class definition 9−13
class example 9−13
class, friend 11−7
class member function 9−13
class, member function in 9−6
class nested class 9−13
class restriction 9−13
class restriction, `static` member 9−10
class, scope of 9−13
object, `static` 3−25
object storage duration 3−25
scope 3−6
`static`, destruction of 6−7
`static` initialization 6−7
variable, destruction of 6−5/6
`<locale>` 22−1
`locale`
`category` 22−4
`classic` 22−9
`combine` 22−8
`facet` 22−6
`global` 22−9
`hasfacet` 22−9
`id` 22−7
`locale()` 22−7
`name` 22−8
`operator!=` 22−8
`operator()` 22−8
`operator==` 22−8
`usefacet` 22−9
`~locale()` 22−8
`locale()`, `locale` 22−7
`~locale()`, `locale` 22−8
`<locale.h>` D−1
locale-specific behavior 1−3
`log` 26−20, 33
`complex` 26−8
`log10` 26−20, 33
`complex` 26−9
logical
AND operator 5−25
AND operator, side effects and 5−25
OR operator 5−26
OR operator, side effects and 5−26
negation operator 5−14/15
`logical_and` 20−10
`logical_not` 20−10
`logical_or` 20−10
`logic_error` 19−1
`logic_error` 19−2
`logic_error` 19−2
`long`
`double` literal 2−11
`double` type 3−32
literal 2−8
type 3−32
type specifier 7−8
type, `unsigned` 3−32
`typedef` and 7−2
`longjmp` 18−21
long-suffix 2−8
look up, name 3−9
lookup
argument-dependent 3−12
member name 10−4
name 3−1
template name 14−24
`lower_bound` 25−20
lowercase 17−6
lvalue 3−34
assignment and 5−27
cast 5−10/11
cast, `reinterpret_cast`, 5−11
cast, `static_cast`, 5−10
conversion to rvalue 4−2
modifiable 3−34
lvalue-to-rvalue conversion 4−2

M

macro
definition scope 16−6
function-like 16−4
invocation 16−5
masking 17−13

name 16−5
object-like 16−4
parameters 16−5
preprocessor 16−1
replacement 16−4
main() 3−22
implementation-defined linkage of 3−22
implementation-defined parameters to 3−22
parameters to 3−22
return from 3−22, 24
make_heap 25−25
make_pair 20−6
malloc 20−20/C−13
<map> 23−29
map 23−31
operator< 23−34
operator== 23−34
operator[] 23−34
mask_array 26−26
fill 26−27
mask_array 26−27
mask_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
masking macro 17−13
<math.h> D−1
max 25−26
valarray 26−17
max_element 25−26
max_length, codecvt 22−19
max_size, basic_string 21−15
meaning of declarator 8−4
member
—see also base class member
access operator, overloaded 13−22
access ambiguity 10−4
access, base class 10−1
access, class 5−6
access, struct default 9−1
access, union default 9−1
array 9−4
cast, pointer to 5−11/12
class object 9−4
constructor order of execution 12−2
declaration 9−3
declaration, class 9−3
declaration, static 3−1
definition, static 9−9
destructor order of execution 12−8
enumerator 7−11
example, static 9−9
function and access control 12−1
function and friend function 11−5
function call, undefined 9−6
function, class 9−5
function, const 9−7/8
function, constructor and 12−2
function definition 9−5
function, destructor and 12−8
function example 9−7, 11−5
function, friend 11−6
function in local class 9−6
function, inline 9−5
function, local class 9−13
function, nested class 11−10
function, overload resolution and 13−5
function, static 9−8/9
function template 14−12
function, union 9−10
function, virtual 17−11, 14
function, volatile 9−7
initialization 12−13
initialization, const 12−14
initialization, order of 12−14
initialization, reference 12−14
initialization, static 9−9
initializer 8−15
initializer, scope of 12−15
local class restriction, static 9−10
name access 11−1
name access example 11−5
name lookup 10−4
name, overloaded 9−4
object initialization 12−14
pointer to —see pointer to member
pointer value, null 4−5
static 9−8
static class 3−25
storage duration, class 3−27
template and static 14−13
type of static 5−14
use, static 9−8
member-declaration 9−3
member-declarator 9−3
member-specification 9−3
memchr 21−32
mem_fun 20−13
mem_fun1 20−13
mem_fun1_ref 20−14
mem_fun1_ref_t 20−14
mem_fun1_t 20−13
mem_fun_ref 20−14
mem_fun_ref_t 20−13
mem_fun_t 20−13
mem-initializer 12−13
mem-initializer-id 12−13
memory
management —see also new, delete
model 1−4
<memory> 20−14
merge 25−22
list 23−20
message, diagnostic 1−2
messages 22−43
close 22−44
do_close 22−45
do_get 22−44
do_open 22−44
open 22−44
messages_byname 22−45
min 25−26
valarray 26−17
min_element 25−26
minus 20−9
mismatch 25−11
missing storage class specifier 7−3
mixed pointer and pointer to member type, multi-level 4−3
mod 26−33
modf 26−33
modifiable lvalue 3−34
modifier function 17−2
modulus
implementation defined 5−22

operator 5−21
zero, undefined 5−1
modulus 20−9
money_get 22−37
do_get 22−38
get 22−38
moneypunct 22−41
do_curr_symbol 22−42
do_decimal_point 22−42
do_frac_digits 22−43
do_grouping 22−42
do_neg_format 22−43
do_negative_sign 22−43
do_pos_format 22−43
do_positive_sign 22−43
do_thousands_sep 22−42
moneypunct_byname 22−43
money_put 22−39
do_put 22−40
put 22−40
most
derived class 1−5
derived object 1−5
multibyte
character 1−3
encoding 2−12
string, null-terminated 17−7
multicharacter
literal 2−9
literal, implementation-defined value of 2−9
multidimensional
array 8−9
array declarator 8−8
multi-level
mixed pointer and pointer to member type 4−3
pointer to member type 4−3
multimap 23−35
operator< 23−37
operator== 23−37
multiple
declaration 3−22
inheritance 10−1/2
inheritance DAG 10−3
inheritance, virtual and 10−8
multiplication operator 5−21
multiplicative operator 5−21
multiplicative-expression 5−21
multiset 23−40
operator< 23−42
operator== 23−42
mutable 7−3

N

name 2−6, 3−1, 5−3
address of cv-qualified 5−14
and translation unit 3−1
class —see class name
declaration 3−1
dependent 14−29, 32
elaborated enum 7−9
global 3−7
hiding 3−5, 8, 5−3, 6−6
hiding, class definition 9−2
hiding, function 13−3
hiding, overloading versus 13−3
hiding, user-defined conversion and 12−6
length of 2−6
look up 3−9
lookup 3−1
lookup, member 10−4
lookup, template 14−24
macro 16−5
overloaded function 13−1
overloaded member 9−4
point of declaration 3−5
qualified 3−13
reserved 17−10
resolution, template 14−24
scope of 3−4
space, label 6−1
unqualified 3−9
name
locale 22−8
type_info 18−15
namespace 17−8/D−1
definition 7−12
global 17−11
scope 3−6
scope, anonymous union at 9−10
scope, global 3−7
namespaces 7−12
narrow string literal 2−11
narrow
basic_ios 27−18
ctype 22−12
ctype<char> 22−16
narrow-character literal 2−9
NDEBUG 17−9
negate 20−9
negation operator, logical 5−14/15
nested
class definition 9−12
class definition example 9−12, 11−10
class example 9−12
class forward declaration example 9−12
class friend function 9−13
class, local class 9−13
class member function 11−10
class, scope of 9−12
type name 9−13
type name example 9−13
type name, scope of 9−13
nested-name-specifier 5−3
<new> 17−12, 18−10
new 3−25, 5−16/17, 12−9
array 5−16
array of class objects and 5−18
constructor and 5−18
default constructor and 5−18
exception and 5−18
initialization and 5−18
operator 17−12, 18−11, 13, 20−20
placement syntax 5−17
scoping and 5−16
storage allocation 5−16
type of 12−10
unspecified constructor and 5−18
unspecified order of evaluation 5−18
new[], operator 17−12, 18−12/13
new-declarator 5−16
new-expression 5−16
new_handler 3−26, 17−12, 18−14
new-initializer 5−16
new-line 2−9
new-placement 5−16
new-type-id 5−16
next_permutation 25−27
noboolalpha 27−20
nondeduced context 14−49
nondigit 2−6

none, bitset 23−47
non-trivial
 constructor 12−2
 destructor 12−7
nonvirtual base class DAG 10−3
nonzero-digit 2−7
norm, complex 26−8
noshowbase 27−20
noshowpoint 27−20
noshowpos 27−21
noskipws 27−21
not1 20−11
not2 20−11
notation, syntax 1−4
not_equal_to 20−10
nounitbuf 27−21
nouppercase 27−21
NTBS 17−6/7, 27−61/D−11
 static 17−7
nth_element 25−20
NTMBS 17−7
 static 17−7
NTWCS 17−7
 static 17−7
null
 character 0 2−11
 directive 16−9
 member pointer value 4−5
 pointer constant 4−4/5
 pointer value 4−4
 reference 8−7
 statement 6−1
NULL 18−2
null-terminated
 byte string 17−6
 multibyte string 17−7
 wide-character string 17−7
number
 hex 2−10
 octal 2−10
numeric type requirements 26−1
<numeric> 26−29
numeric_limits 3−33, 18−3
num_get 22−21
 do_get 22−23
 get 22−22
numpunct 22−29
 decimal_point 22−30
 do_decimal_point 22−31
 do_grouping 22−31
 do_thousands_sep 22−31
 do_truename do_falsename 22−31
 grouping 22−30
 thousands_sep 22−30
 truename falsename 22−30
numpunct_byname 22−31
num_put 22−25
 do_put 22−26
 put 22−26

O

object 1−4, 3−1, 34
 class —see also class object
 complete 1−4
 definition 3−2
 delete 5−19
 destructor and placement of 12−9
 destructor static 3−24
 initialization, auto 8−15
 initialization, static 3−23, 8−15/16
 layout, access specifier and 11−3
 lifetime 3−27
 linkage, implementation-defined 7−26
 linkage specification 7−26
 representation 3−30
 state 17−2
 static local 3−25
 storage duration, local 3−25
 temporary —see temporary
 type 1−4
 type 3−31
 type, completely defined 9−4
 undefined deleted 3−27
 unnamed 12−2
object-expression 5−1
object-like macro 16−4
observer function 17−2
oct 27−22
octal
 literal 2−8
 number 2−10
octal-escape-sequence 2−9
octal-literal 2−7
of
 overloading, example 13−1
 reference, direct binding 8−22
offsetof 18−2/C−13
ofstream 27−4, 59
old function, plain 18−21
one-definition rule 3−2
one's complement operator 5−14/15
open
 basic_filebuf 27−61, 68
 basic_ifstream 27−65
 basic_ofstream 27−67
 messages 22−44
openmode, ios_base 27−11
operations on class object 9−1
operator
 —see conversion function, conversion
 %= 5−27
 &= 5−27
 *= 5−27
 += 5−15, 27
 -= 5−27
 /= 5−27
 <<= 5−27
 >>= 5−27
 ^= 5−27
 additive 5−22
 address-of 5−14
 assignment 5−27, 17−7
 bitwise 5−25
 bitwise AND 5−25
 bitwise exclusive OR 5−25
 bitwise inclusive OR 5−25
 cast 5−13, 20, 8−2
 class member access 5−6
 comma 5−28
 conditional expression 5−26
 copy assignment 12−19
 decrement 5−8, 14/15
 division 5−21
 equality 5−24
 example, scope resolution 10−5
 function call 5−5, 13−20
 function call 13−20
 greater than 5−23
 greater than or equal to 5−23
 identities and overloading 13−20

increment 5−8, 14/15
indirection 5−14
inequality 5−24
left shift —see left shift operator
less than 5−23
less than or equal to 5−23
list 2−7, 13−19
logical AND 5−25
logical OR 5−26
logical negation 5−14/15
modulus 5−21
multiplication 5−21
multiplicative 5−21
new —see new
one's complement 5−14/15
overloaded 5−1
overloading —see also overloaded operator
overloading restrictions 13−20
pointer to member 5−21
precedence of 1−6
relational 5−23
right shift; right shift operator 5−23
scope resolution 5−3, 9−6, 10−1, 10
shift —see left shift operator, right shift operator
side effects and comma 5−28
side effects and logical AND 5−25
side effects and logical OR 5−26
sizeof 5−13, 15
subscripting 5−4, 13−20
unary 5−13/14
unary minus 5−14
unary plus 5−14
use, scope resolution 9−9
|= 5−27
operator
bool, basic_ios 27−19
bool(), basic_istream 27−35
bool(), basic_ostream 27−44
delete 17−12, 18−12, 20−20
delete 5−19, 12−10
delete —see delete
delete[] 17−12, 18−13
delete[] 5−19, 12−10
function 13−19
new 17−12, 18−11, 13, 20−20
new 5−17, 12−9
new[] 17−12, 18−12/13
new[] 5−17, 12−9
overloaded 13−19
operator!
basic_ios 27−19
valarray 26−16
operator!= 20−5
basic_string 21−27
bitset 23−47
complex 26−7
istreambuf_iterator 24−24
locale 22−8
reverse_iterator 24−14
type_info 18−15
valarray 26−19
operator%, valarray 26−18
operator%=
gslice_array 26−26
indirect_array 26−29
mask_array 26−27
slice_array 26−23
valarray 26−16
operator&
bitset 23−47
valarray 26−18
operator&&, valarray 26−18/19
operator&=
bitset 23−45
gslice_array 26−26
indirect_array 26−29
mask_array 26−27
slice_array 26−23
valarray 26−16
operator(), locale 22−8
operator*
auto_ptr 20−20
back_insert_iterator 24−16
complex 26−7
front_insert_iterator 24−17
insert_iterator 24−18
istreambuf_iterator 24−23
ostreambuf_iterator 24−25
reverse_iterator 24−13
valarray 26−18
operator*=
complex 26−6
gslice_array 26−26
indirect_array 26−29
mask_array 26−27
slice_array 26−23
valarray 26−16
operator+
basic_string 21−26
complex 26−6
reverse_iterator 24−13, 15
valarray 26−16, 18
operator++
back_insert_iterator 24−16
front_insert_iterator 24−17
insert_iterator 24−18
istreambuf_iterator 24−24
ostreambuf_iterator 24−25
reverse_iterator 24−13
operator+=
basic_string 21−17
complex 26−6
gslice_array 26−26
indirect_array 26−29
mask_array 26−27
reverse_iterator 24−14
slice_array 26−23
valarray 26−16
operator-
complex 26−6
reverse_iterator 24−14/15
valarray 26−16, 18
operator--, reverse_iterator 24−13
operator-=
complex 26−6
gslice_array 26−26
indirect_array 26−29
mask_array 26−27
reverse_iterator 24−14
slice_array 26−23
valarray 26−16
operator->
auto_ptr 20−20
reverse_iterator 24−13
operator/, valarray 26−18
operator/=
complex 26−6
gslice_array 26−26
indirect_array 26−29
mask_array 26−27
slice_array 26−23
valarray 26−16

operator<
 basic_string 21−27
 map 23−34
 multimap 23−37
 multiset 23−42
 pair 20−6
 queue 23−22
 reverse_iterator 24−14
 set 23−40
 valarray 26−19
 vector 23−26
 vector<bool> 23−29
operator<< 27−22
 basic_ostream 27−45/46
 basic_string 21−28
 bitset 23−47/48
 complex 26−7
 valarray 26−18
operator<<=
 bitset 23−45
 gslice_array 26−26
 indirect_array 26−29
 mask_array 26−27
 slice_array 26−23
 valarray 26−16
operator<= 20−6
 basic_string 21−28
 reverse_iterator 24−15
 valarray 26−19
operator=
 auto_ptr 20−19
 back_insert_iterator 24−16
 bad_alloc 18−14
 bad_cast 18−16
 bad_exception 18−18
 bad_typeid 18−16
 basic_string 21−14
 exception 18−17
 front_insert_iterator 24−17
 gslice_array 26−25
 indirect_array 26−28
 insert_iterator 24−18
 mask_array 26−27
 ostreambuf_iterator 24−25
 slice_array 26−22
 type_info 18−15
 valarray 26−14/15
operator==
 basic_string 21−26
 bitset 23−47
 complex 26−7
 istream_iterator 24−21
 istreambuf_iterator 24−24
 locale 22−8
 map 23−34
 multimap 23−37
 multiset 23−42
 pair 20−6
 queue 23−22
 reverse_iterator 24−14
 set 23−40
 type_info 18−15
 valarray 26−19
 vector 23−26
 vector<bool> 23−29
operator> 20−5
 basic_string 21−27
 reverse_iterator 24−14
 valarray 26−19
operator>= 20−6
 basic_string 21−28
 reverse_iterator 24−15
 valarray 26−19
operator>>
 basic_istream 27−36
 basic_string 21−28
 bitset 23−47/48
 complex 26−7
 istream 27−35
 valarray 26−18
operator>>=
 bitset 23−46
 gslice_array 26−26
 indirect_array 26−29
 mask_array 26−27
 slice_array 26−23
 valarray 26−16
operator[]
 basic_string 21−16
 map 23−34
 reverse_iterator 24−14
 valarray 26−15
operator^
 bitset 23−48
 valarray 26−18
operator^=
 bitset 23−45
 gslice_array 26−26
 indirect_array 26−29
 mask_array 26−27
 slice_array 26−23
 valarray 26−16
operator|
 bitset 23−47
 valarray 26−18
operator|=
 bitset 23−45
 gslice_array 26−26
 indirect_array 26−29
 mask_array 26−27
 slice_array 26−23
 valarray 26−16
operator||, valarray 26−18/19
operator~
 bitset 23−46
 valarray 26−16
operator 13−19
operator-function-id 13−19
optimization of temporary —see elimination of temporary
OR
 operator, bitwise exclusive 5−25
 operator, bitwise inclusive 5−25
 operator, logical 5−26
 operator, side effects and logical 5−26
order
 of argument evaluation 5−6
 of argument evaluation, unspecified 5−6
 of base class initialization 12−14
 of destruction of temporary 12−4
 of evaluation new, unspecified 5−18
 of evaluation of expression 1−6
 of evaluation of function arguments, implementation-defined 8−13
 of evaluation, unspecified 3−23, 5−1
 of execution, base class constructor 12−2
 of execution, base class destructor 12−8
 of execution, constructor and array 12−12
 of execution, constructor and static objects 12−13
 of execution, destructor 12−8
 of execution, destructor and array 12−8
 of execution, member constructor 12−2
 of execution, member destructor 12−8

of function call evaluation, unspecified 5−6
of initialization 3−23, 10−2
of member initialization 12−14
of virtual base class initialization 12−14
ordering, function template partial 14−22
<ostream> 27−31
ostream 27−4, 31
ostreambuf_iterator 24−24
failed 24−25
operator* 24−25
operator++ 24−25
operator= 24−25
ostreambuf_iterator 24−25
ostreambuf_iterator 24−25
ostream_iterator 24−21
ostringstream 27−4, 51
ostrstream D−10
:pcount D−11
freeze D−11
ostrstream D−11
ostrstream D−11
rdbuf D−11
str D−11
out, codecvt 22−18
out_of_range 19−3, 21−9, 23−44/47
out_of_range 19−3
out_of_range 19−3
output_iterator_tag 24−9/10
overflow 5−1
undefined 5−1
overflow
basic_filebuf 27−62
basic_streambuf 27−30
basic_stringbuf 27−53
strstreambuf D−7
overflow_error 19−4, 23−44, 47
overflow_error 19−4
overflow_error 19−4
overload
resolution 13−4
resolution and conversion 13−12
resolution and default argument 13−11
resolution and ellipsis 13−11
resolution and member function 13−5
resolution and pointer conversion 13−19
resolution contexts 13−4
resolution, function template 14−53
resolution, template 14−22
overloaded
assignment and initialization 12−12
assignment operator 13−21
binary operator 13−21
declaration 13−1
decrement operator 13−22
function, address of 5−14, 13−18
function ambiguity detection 13−4
function call operator 13−21
function call resolution —see also argument matching, overload resolution
function declaration matching 13−3
function, linkage specification 7−25
function name 13−1
increment operator 13−22
member access operator 13−22
member name 9−4
name and friend declaration 11−6
operator 13−19
operator 5−1
operator 13−19
operator and default argument 13−20
operator, inheritance of 13−20
subscripting operator 13−21
unary operator 13−20
overloading 8−10, 9−2, 13−1, 14−21
and access 13−4
and const 13−2
and default initializers 13−3
and delete 3−27
and derived class 13−3
and enum 13−2
and equivalent parameter declarations 13−2
and pointer versus array 13−2
and return type 13−1
and scope 13−3
and static 13−1
and typedef 13−2
and volatile 13−2
operator identities and 13−20
postfix ++ and -- 13−22
prefix ++ and -- 13−22
resolution and access control 10−5
restriction 13−20
subsequence rule 13−16
versus name hiding 13−3
overrider, final 10−7

P

pair 20−6
operator< 20−6
operator== 20−6
parameter 1−3
adjustment, array 8−10
adjustment, function 8−10
declaration 8−9
example, unnamed 8−15
initialization 5−5
list example, variable 8−9
list, variable 5−5, 8−9
reference 8−6
scope of 3−6
void 8−9
parameter type list 8−10
parameter-declaration 8−9
parameterized type —see template
parameters
macro 16−5
to main() 3−22
to main(), implementation-defined 3−22
parentheses
and ambiguity 5−16
in declaration 8−3, 5
parenthesized expression 5−3
partial
ordering, function template 14−22
specializations, class template 14−16
partial_sort 25−19
partial_sort_copy 25−20
partial_sum 26−31
partition 25−18
pbackfail
basic_filebuf 27−62
basic_streambuf 27−29
basic_stringbuf 27−53
strstreambuf D−7
pbase, basic_streambuf 27−27
pbump, basic_streambuf 27−27
:pcount, ostrstream D−11
pcount
strstream D−13
strstreambuf D−6

peek, basic_istream 27−40
period 17−6
phases, translation 2−1
placement
  of object, destructor and 12−9
  syntax, new 5−17
plain old function 18−21
plus 20−9
pm-expression 5−21
POD
  class type 5−18
  type 3−31
  type 5−18
POD-struct 9−1
POF 18−21
point
  of declaration class name 9−3
  of declaration, enumerator 3−5
  of declaration name 3−5
  of definition, enumerator 7−10
  of error checking 14−25
  of instantiation 14−32
  promotion, floating 4−4
  type, floating 3−32
pointer
  —see also void*
  and pointer to member type, multi-level mixed 4−3
  arithmetic 5−22
  cast, integer to 5−11
  comparison 5−24
  comparison, undefined 5−22, 24
  comparison, unspecified 5−24
  comparison, void* 5−24
  constant, null 4−4/5
  conversion 4−4
  conversion, array 4−2
  conversion, overload resolution and 13−19
  declaration 8−5
  declarator * 8−5
  example, constant 8−5
  integer conversion, implementation defined 5−11
  subtraction, implementation defined 5−22
  terminology 3−33
  to abstract class 10−10
  to bit-field restriction 9−11
  to function cast 5−11
  to function cast, undefined 5−11
  to function comparison 5−24
  to function conversion 4−2
  to integer cast 5−11
  to member 3−33, 5−21
  to member cast 5−11/12
  to member constant expression 5−14
  to member conversion 4−5
  to member declarator ::* 8−7
  to member example 8−7
  to member function 5−21
  to member operator 5−21
  to member type, multi-level 4−3
  to member type, multi-level mixed pointer and 4−3
  to member void* conversion 4−5
  type 3−33
  value, null 4−4
  value, null member 4−5
  versus array, overloading and 13−2
  zero 4−4
pointer_to_binary_function 20−13
pointer_to_unary_function 20−12
polar, complex 26−8
polymorphic
  class 10−6
  type 10−6
pop, priority_queue 23−23
pop_heap 25−25
postfix
  ++ and -- 5−8
  ++ and --, overloading 13−22
  expression 5−4
potential scope 3−4
pow 26−20, 33
  complex 26−9
pp-number 2−5
pptr, basic_streambuf 27−27
pragma directive 16−9
#pragma 16−9
precedence of operator 1−6
precision, ios_base 22−10, 27−13
prefix
  ++ and -- 5−15
  ++ and --, overloading 13−22
  L 2−9, 11
preprocessing 16−1
  directive 16−1
preprocessing-op-or-punc 2−7
preprocessing-token 2−3
preprocessor, macro 16−1
prev_permutation 25−27
primary
  expression 5−2
  template 14−16
priority_queue 23−22
  pop 23−23
  priority_queue 23−23
  priority_queue 23−23
  push 23−23
private 11−1
  base class 11−3
program 3−20
  environment 3−22
  ill-formed 1−3
  start 3−22/23
  startup 17−10, 12
  termination 3−22, 24
  termination and destructor 12−8
  well-formed 1−4
promotion
  floating point 4−4
  integral 4−3
  to int, bool 4−3
protected 11−1
protection 17−14
  —see access control
proxy, istreambuf_iterator 24−23
pseudo destructor call 5−6
pseudo-destructor-name 5−6
pseudo-destructor-name 5−4
ptrdiff_t 5−22
  implementation defined type of 5−22
ptr_fun 20−12/13
ptr-operator 8−1
pubimbue, basic_streambuf 27−25
public 11−1
  base class 11−3
pubseekoff, basic_streambuf 27−25
pubseekpos, basic_streambuf 27−26
pubsetbuf, basic_streambuf 27−25
pubsync, basic_streambuf 27−26
punctuators 2−7
pure
  specifier 9−4
  virtual destructor 12−8
  virtual function 10−10/11

virtual function call, undefined 10−11
virtual function definition 10−10
virtual function example 10−10
pure-specifier 9−4
`push`, `priority_queue` 23−23
`push_heap` 25−25
`put`
`basic_ostream` 27−47
`money_put` 22−40
`num_put` 22−26
`time_put` 22−36
`putback`, `basic_istream` 27−40
`pword`, `ios_base` 27−14

Q

qualification
conversions 4−2
explicit 3−13
qualified
id 5−3
name 3−13
qualified-id 5−3
question mark 2−9
`<queue>` 23−10
`queue` 23−21
`operator<` 23−22
`operator==` 23−22
quote
double 2−9
single 2−9

R

`random_access_iterator_tag` 24−9/10
`random_shuffle` 25−17
`range_error` 19−3
`range_error` 19−3
`range_error` 19−3
rank, conversion 13−14
`rbegin`, `basic_string` 21−15
`rdbuf`
`basic_filebuf` 27−68
`basic_ifstream` 27−65
`basic_ios` 27−18
`basic_istringstream` 27−56
`basic_ofstream` 27−66
`basic_ostringstream` 27−57
`basic_stringstream` 27−58
`istrstream` D−10
`ostrstream` D−11
`strstream` D−12
`rdstate`, `basic_ios` 27−19
`read`, `basic_istream` 27−40
`readsome`, `basic_istream` 27−40
`real` 26−7
`complex` 26−8
`realloc` 20−20
recursive function call 5−6
redefinition
enumerator 7−10
`typedef` 7−5
reentrancy 17−14
implementation-defined behavior 17−14
reference 3−33
and argument passing 8−21
and `return` 8−21
argument 5−5
assignment 8−21
assignment to 5−27
binding 8−21
call by 5−5
cast 5−10, 12
cast, `reinterpret_cast`, 5−12
cast, `static_cast`, 5−10
`const` 8−22
declaration 8−6
declaration, `extern` 8−21
declarator `&` 8−6
direct binding of 8−22
expression 5−1
initialization 8−7, 21
member initialization 12−14
null 8−7
parameter 8−6
restriction 8−7
`sizeof` 5−15
reference-compatible 8−21
reference-related 8−21
region, declarative 3−1, 4
`register` 7−3
declaration 7−3
restriction 7−3
`register_callback`, `ios_base` 27−14
reinterpret cast 5−11
`reinterpret_cast`
lvalue cast 5−11
reference cast 5−12
relational operator 5−23
relational-expression 5−23
`release`, `auto_ptr` 20−20
remainder operator —see modulus operator
`remove` 25−15
`list` 23−20
`remove_copy` 25−15
`remove_copy_if` 25−15
`remove_if` 25−15
`rend`, `basic_string` 21−15
`replace` 25−14
`basic_string` 21−20
`replace_copy` 25−14
`replace_copy_if` 25−14
`replace_if` 25−14
replacement
function 17−2
macro 16−4
representation
object 3−30
value 3−30
required behavior 17−2, 5
requirements 17−3
`Allocator` 20−2
`Assignable` 23−1
`CopyConstructible` 20−2
`EqualityComparable` 20−1
`LessThanComparable` 20−1
container 23−1
iterator 24−1
numeric type 26−1
reraise 15−3
rescanning and replacement 16−6
`reserve`
`basic_string` 21−16
`vector` 23−26
reserved
function 17−2
identifier 2−6
name 17−10
word —see keyword
`reset`, `bitset` 23−46
`resetiosflags` 27−48

resize
basic_string 21−16
deque 23−15
list 23−18
valarray 26−18
vector 23−27
resolution
and conversion, overload 13−12
and default argument, overload 13−11
and ellipsis, overload 13−11
and member function, overload 13−5
and pointer conversion, overload 13−19
argument matching —see overload
function template overload 14−53
overload 13−4
overloaded function call resolution —see also argument matching, overload
overloading —see overload resolution
resolution overloading —see overload
scoping ambiguity 10−5
template name 14−24
template overload 14−22
restriction 17−13/14
address of bit-field 9−11
anonymous union 9−10
auto 7−3
bit-field 9−11
constructor 12−1/2
copy assignment operator 12−22
copy constructor 12−20
destructor 12−7/8
enumerator 7−10
extern 7−3
local class 9−13
overloading 13−20
pointer to bit-field 9−11
reference 8−7
register 7−3
static 7−3
static member local class 9−10
union 9−10, 12−2
restrictions, operator overloading 13−20
rethrow 15−3
return
type 8−10
type conversion 6−6
type, overloading and 13−1
return 6−5/6
constructor and 6−6
from main() 3−22, 24
reference and 8−21
statement —see also return
reverse 25−16
list 23−20
reverse_copy 25−16
reverse_iterator 24−11
conversion 24−13
operator!= 24−14
operator* 24−13
operator+ 24−13, 15
operator++ 24−13
operator+= 24−14
operator- 24−14/15
operator-- 24−13
operator-= 24−14
operator-> 24−13
operator< 24−14
operator<= 24−15
operator== 24−14
operator> 24−14
operator>= 24−15
operator[] 24−14
reverse_iterator 24−12
reverse_iterator 24−12
rfind, basic_string 21−22
right
shift, implementation defined 5−23
shift operator 5−23
right 27−21
rotate 25−17
rotate_copy 25−17
rounding 4−4
rule
as-if 1−5
one-definition 3−2
rules
for *condition*s 6−2
summary, scope 3−9
run-time initialization 3−23
runtime_error 19−3
runtime_error 19−3
runtime_error 19−3
rvalue 3−34
lvalue conversion to 4−2

S

sbumpc, basic_streambuf 27−26
scalar type 3−31
scan_is
ctype 22−11
ctype<char> 22−15
scan_not
ctype 22−11
ctype<char> 22−16
s-char 2−11
s-char-sequence 2−11
scientific 27−22
scope 3−1, 4
anonymous union at namespace 9−10
class 3−7
destructor and exit from 6−5
exception declaration 3−6
function 3−6
function prototype 3−6
global 3−7
global namespace 3−7
iteration-statement 6−4
local 3−6
macro definition 16−6
namespace 3−6
of class definition 9−2
of class name 9−2
of declaration in for 6−5
of default argument 8−13
of delete example 12−11
of enumerator class 7−11
of label 3−6, 6−1
of local class 9−13
of member initializer 12−15
of name 3−4
of nested class 9−12
of nested type name 9−13
of parameter 3−6
overloading and 13−3
potential 3−4
resolution operator 5−3, 9−6, 10−1, 10
resolution operator :: 3−13
resolution operator example 10−5
resolution operator use 9−9
rules summary 3−9

selection-statement 6–2
scoping
ambiguity resolution 10–5
and `new` 5–16
`search` 25–12
`seekdir`, `ios_base` 27–11
`seekg`, `basic_istream` 27–40
`seekoff`
`basic_filebuf` 27–63
`basic_streambuf` 27–28
`basic_stringbuf` 27–53
`strstreambuf` D–8
`seekp`, `basic_ostream` 27–44
`seekpos`
`basic_filebuf` 27–64
`basic_streambuf` 27–28
`basic_stringbuf` 27–54
`strstreambuf` D–9
selection statement 6–2
selection-statement 6–2
scope 6–2
semantics, class member 5–6
`sentry`
`basic_istream` 27–34
`basic_ostream` 27–44
`~sentry`
`basic_istream` 27–35
`basic_ostream` 27–44
separate
compilation 2–1
translation 2–1
sequence
implicit conversion 13–13
point 1–5, 5–1
standard conversion 4–1
statement 6–1
sequencing operator —see comma operator
set, basic source character 2–2
`<set>` 23–30
`set` 23–38
`bitset` 23–46
`operator<` 23–40
`operator==` 23–40
`setbase` 27–49
`setbuf`
`basic_filebuf` 27–63
`basic_streambuf` 27–28
`streambuf` D–9
`strstreambuf` D–9
`set_difference` 25–24
`setf`, `ios_base` 27–12
`setfill` 27–49
`setg`, `basic_streambuf` 27–27
`set_intersection` 25–23
`setiosflags` 27–49
`setjmp` 17–11
`<setjmp.h>` D–1
`setlocale` 17–6
`set_new_handler` 17–12, 18–14
`setp`, `basic_streambuf` 27–27
`setprecision` 27–49
`setstate`, `basic_ios` 27–19
`set_symmetric_difference` 25–24
`set_terminate` 17–12, 18–19
`set_unexpected` 17–12, 18–19
`set_union` 25–23
`setw` 27–50
`sgetc`, `basic_streambuf` 27–26
`sgetn`, `basic_streambuf` 27–26
shift operator —see left shift operator, right shift operator
`shift`, `valarray` 26–17
shift-expression 5–23
`short`
type 3–32
type specifier 7–8
type, `unsigned` 3–32
`typedef` and 7–2
`showbase` 27–20
`showmanyc`
`basic_filebuf` 27–62
`basic_streambuf` 27–28, 62
`showpoint` 27–20
`showpos` 27–20
side
effect 1–5
effects 5–1
effects and comma operator 5–28
effects and logical AND operator 5–25
effects and logical OR operator 5–26
sign
of bit-field, implementation-defined 9–11
of `char`, implementation-defined 3–32
sign 2–10
`<signal.h>` D–1
signature 1–3
`signed`
`char` type 3–32
character 3–32
`typedef` and 7–2
`unsigned` integer conversion 4–4
simple-escape-sequence 2–9
simple-type-specifier 7–8
`sin` 26–20, 33
`complex` 26–9
single quote 2–9
`sinh` 26–20, 33
`complex` 26–9
`size`
`basic_string` 21–15
`bitset` 23–47
`gslice` 26–24
`slice` 26–21
`sizeof`
array 5–15
class object 5–15
empty class 9–1
integral type, implementation-defined 3–32
operator 5–13, 15
reference 5–15
string 2–12
type, implementation-defined 3–32
`size_t` 5–15
`skipws` 27–21
`slice` 26–21
`size` 26–21
`slice` 26–21
`slice` 26–21
`start` 26–21
`stride` 26–21
`slice_array` 26–21
`fill` 26–23
`operator%=` 26–23
`operator&=` 26–23
`operator*=` 26–23
`operator+=` 26–23
`operator-=` 26–23
`operator/=` 26–23
`operator<<=` 26–23
`operator=` 26–22
`operator>>=` 26–23
`operator^=` 26–23
`operator|=` 26–23

slice_array 26−22
slice_array 26−22
smanip, implementation-specified 27−48
snextc, basic_streambuf 27−26
sort 25−19
list 23−20
sort_heap 25−25
source
character set, basic 2−2
file 2−1, 17−9, 11
file inclusion 16−3
space, white 2−3
special member function —see also constructor, destructor, inline function, user-defined conversion, virtual function
specialization
class template 14−5
template 14−34
template explicit 14−39
specializations, class template partial 14−16
specification, template argument 14−45
specifications
C++ Standard Library exception 17−15
Standard C library exception 17−15
implementation-defined exception 17−15
specifier
access —see access specifier
auto 7−3
declaration 7−2
explicit 7−5
friend 7−6
friend 17−14
function 7−4
inline 7−5
missing storage class 7−3
static 7−3
storage class 7−3
type —see type specifier
typedef 7−5
virtual 7−5
splice, list 23−19
sputbackc, basic_streambuf 27−26
sputc, basic_streambuf 27−26
sputn, basic_streambuf 27−26
sqrt 26−20, 33
complex 26−9
<sstream> 27−50
stable_partition 25−18
stable_sort 25−19
stack unwinding 15−4
<stack> 23−10
stack 23−23
Standard
C library 17−1, 6, 8, 17−10/C−10, C−12
C library exception specifications 17−15
Library, C++ 17−1, 11, 13/14
library, C++ 17−11
standard
conversion 4−1
conversion sequence 4−1
start, program 3−22/23
start
gslice 26−24
slice 26−21
startup, program 17−10, 12
state, object 17−2
state, fpos 27−15
statement 6−1
—see also return, return
break 6−5/6
compound 6−1
continue 6−5/6
continue in for 6−5
declaration 6−6
declaration in for 6−5
declaration in switch 6−3
do 6−3, 5
empty 6−1
expression 6−1
for 6−3, 5
goto 6−1, 5/6
if 6−2/3
iteration 6−3
jump 6−5
labeled 6−1
null 6−1
selection 6−2
sequence 6−1
switch 6−2/3, 6
while 6−3/4
{}, block 6−1
statement 6−1
static
NTBS 17−7
NTMBS 17−7
NTWCS 17−7
cast 5−10
type 1−3
static 7−3
class member 3−25
data member 9−8
destruction of local 6−7
initialization, local 6−7
linkage of 3−20, 7−3
local object 3−25
member 9−8
member declaration 3−1
member definition 9−9
member example 9−9
member function 9−8/9
member initialization 9−9
member local class restriction 9−10
member, template and 14−13
member, type of 5−14
member use 9−8
object, destructor 3−24
object initialization 3−23, 8−15/16
objects order of execution, constructor and 12−13
overloading and 13−1
restriction 7−3
specifier 7−3
static_cast
conversion to enumeration type 5−11
lvalue cast 5−10
reference cast 5−10
<stdarg.h> D−1
__STDC__ 16−9
implementation-defined 16−9
<stddef.h> 2−9, 2−11/D−1
<stdexcept> 19−1
<stdio.h> D−1
<stdlib.h> D−1
storage
allocation new 5−16
class 3−1
class declaration 7−3
class specifier 7−3
class specifier, missing 7−3
duration 3−24
duration, auto 3−25
duration, class member 3−27
duration, dynamic 3−25, 5−16
duration, local object 3−25

management —see new, delete
of array 8−9
str
basic_istringstream 27−56
basic_ostringstream 27−57
basic_stringbuf 27−52
basic_stringstream 27−58
istrstream D−10
ostrstream D−11
strstream D−13
strstreambuf D−6
strchr 21−31
<streambuf> 27−22
streambuf 27−4, 22
implementation-defined 27−2
setbuf D−9
streamoff 27−15/D−2
implementation-defined 27−7/D−2
streampos, implementation-defined D−2
streamsize 27−7
strftime 22−37
stride
gslice 26−24
slice 26−21
string
concatenation 2−11
distinct 2−11
literal 2−11
literal concatenation, undefined 2−11
literal, implementation-defined 2−11
literal, narrow 2−11
literal, type of 2−11
literal, undefined change to 2−11
literal, wide 2−11
null-terminated byte 17−6
null-terminated multibyte 17−7
null-terminated wide-character 17−7
sizeof 2−12
terminator 0 2−11
type of 2−11
<string> 21−7
stringbuf 27−4, 51
<string.h> D−1
string-literal 2−11
stringstream 27−4
strlen D−6, 11
strpbrk 21−31
strrchr 21−31
strstr 21−31
strstream D−11
pcount D−13
rdbuf D−12
str D−13
strstream D−12
strstream D−12
~strstream D−12
~strstream, strstream D−12
strstreambuf D−3
freeze D−6
overflow D−7
pbackfail D−7
pcount D−6
seekoff D−8
seekpos D−9
setbuf D−9
str D−6
strstreambuf D−5
strstreambuf D−5
underflow D−8
~strstreambuf D−6
~strstreambuf, strstreambuf D−6
struct
class versus 9−1
default member access 9−1
initialization 8−18
type specifier 9−1
structure 9−1
tag —see class name
sub-object 1−4
lattice —see DAG
subscripting
example 8−8
explanation 8−8
operator 5−4, 13−20
operator, overloaded 13−21
subsequence rule, overloading 13−16
substr, basic_string 21−25
subtraction
implementation defined pointer 5−22
operator 5−22
suffix
E 2−11
F 2−11
L 2−8, 11
U 2−8
f 2−11
l 2−8, 11
u 2−8
sum, valarray 26−17
summary
compatibility with ISO C C−1
scope rules 3−9
syntax A−1
sungetc, basic_streambuf 27−26
swap 25−13
basic_string 21−21, 28
swap_ranges 25−13
switch
statement 6−2/3, 6
statement, declaration in 6−3
sync
basic_filebuf 27−64
basic_istream 27−40
basic_streambuf 27−28
sync_with_stdio, ios_base 27−13
synonym 7−15
type name as 7−5
syntax
checking 14−25
class member 5−6
notation 1−4
summary A−1

T

table, ctype<char> 22−16
tan 26−20, 33
complex 26−9
tanh 26−20, 33
complex 26−9
tellg, basic_istream 27−40
tellp, basic_ostream 27−44
template 14−1
and < 14−4/5
and friend 14−15
and static member 14−13
argument 14−6
argument specification 14−45
class 23−44
definition of 14−1
explicit specialization 14−39

function 14−45
implicit instantiation 14−35
member function 14−12
name lookup 14−24
name resolution 14−24
overload resolution 14−22
overload resolution, function 14−53
partial ordering, function 14−22
partial specializations, class 14−16
primary 14−16
specialization 14−34
specialization, class 14−5
type equivalence 14−10
`template` 14−1
template-argument 14−4
template-argument-list 14−4
template-declaration 14−1
template-id 14−4
template-name 14−4
template-parameter 14−2
template-parameter-list 14−1
temporary 12−3
and declarator initializer 12−4
constructor for 12−3
destruction of 12−3
destructor for 12−3
elimination of 12−3
implementation-defined generation of 12−3
order of destruction of 12−4
`terminate` 3−24, 15−9, 18−10, 18/19
`terminate()` 15−9
`terminate_handler` 17−12, 18−19
termination
and destructor, program 12−8
program 3−22, 24
terminator `0`, string 2−11
terminology, pointer 3−33
`test`, `bitset` 23−47
`this` 5−2
pointer —see `this`
type of 9−7
`thousands_sep`, `numpunct` 22−30
`throw` 15−1
throw-expression in conditional-expression 5−26
throw-expression 15−1
throwing, exception 15−2
`tie`, `basic_ios` 27−18
`time_get` 22−33
`date_order` 22−34
`do_date_order` 22−35
`do_get_date` 22−35
`do_get_monthname` 22−35
`do_get_time` 22−35
`do_get_weekday` 22−35
`do_get_year` 22−35
`get_date` 22−34
`get_monthname` 22−34
`get_time` 22−34
`get_weekday` 22−34
`get_year` 22−35
`time_get_byname` 22−36
`<time.h>` D−1
`time_put` 22−36
`do_put` 22−37
`put` 22−36
`time_put_byname` 22−37
`times` 20−9
to
int, bool promotion 4−3
rvalue, lvalue conversion 4−2
token 2−4, 7
token 2−4
`tolower` 22−10
`ctype` 22−12
`ctype<char>` 22−16
`to_string`, `bitset` 23−47
`to_ulong`, `bitset` 23−46
`toupper` 22−10
`ctype` 22−11
`ctype<char>` 22−16
`transform` 25−14
`collate` 22−32
translation
phases 2−1
separate 2−1
unit 17−9/10
unit 2−1, 3−20
unit, name and 3−1
trigraph 2−1, 3
`truename` `falsename`, `numpunct` 22−30
truncation 4−4
`try` 15−1
try-block 15−1
type 3−1
Boolean 3−32
POD 3−31
ambiguity, declaration 7−2
arithmetic 3−33
array 3−33, 8−10
bitmask 17−5/6
`char` 3−32
character 3−32
checking, argument 5−5
checking of default argument 8−12
class and 9−1
completely defined object 9−4
compound 3−33
`const` 7−6
conversion, explicit —see casting
declaration 8−4
declaration consistency 3−22
declaration, `typedef` as 7−5
definition, class name as 9−2
destination 8−17
`double` 3−32
dynamic 1−2
enumerated 3−33, 17−5
enumeration underlying 7−11
equivalence 7−5, 9−2
equivalence, template 14−10
example of incomplete 3−31
`float` 3−32
floating point 3−32
function 3−33, 8−9/10
fundamental 3−32
generator —see template
implementation-defined `sizeof` 3−32
incomplete 3−2/3, 5, 31, 4−2, 5−4/9, 14/16, 19, 22, 27, 10−1
`int` 3−32
integral 3−32
`long` 3−32
`long double` 3−32
multi-level mixed pointer and pointer to member 4−3
multi-level pointer to member 4−3
name 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
object 1−4

of bit-field 9−11
of character literal 2−9
of constructor 12−2
of conversion 12−6
of `delete` 12−10
of `enum` 7−10
of floating point literal 2−11
of integer literal 2−8
of integer literal, implementation-defined 2−8
of `new` 12−10
of `ptrdiff_t`, implementation defined 5−22
of `static` member 5−14
of string 2−11
of string literal 2−11
of `this` 9−7
pointer 3−33
polymorphic 10−6
pun 5−12
requirements, numeric 26−1
`short` 3−32
`signed char` 3−32
specifier, `char` 7−8
specifier, `class` 9−1
specifier, `double` 7−8
specifier, `enum` 7−9
specifier, `float` 7−8
specifier, `int` 7−8
specifier, `long` 7−8
specifier, `short` 7−8
specifier, `struct` 9−1
specifier, `union` 9−1
specifier, `unsigned` 7−8
specifier, `void` 7−8
specifier, `volatile` 7−8
static 1−3
`unsigned` 3−32
`unsigned char` 3−32
`unsigned int` 3−32
`unsigned long` 3−32
`unsigned short` 3−32
`void` 3−33
`void*` 3−33
`volatile` 7−6
`wchar_t` 3−32
`wchar_t` underlying 3−32
typedef, function 8−10
`typedef`
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 3−1
enum name 7−6
example 7−5
overloading and 13−2
redefinition 7−5
specifier 7−5
typedef-name 7−5
`typeid` 5−9
type-id 8−2
type-id-list 15−6
`<typeinfo>` 18−14
`type_info` 5−9, 18−15
`before` 18−15
`name` 18−15
`operator!=` 18−15
`operator=` 18−15
`operator==` 18−15
`type_info` 18−15
`type_info` 18−15
`type_info::name`, implementation-defined 18−15
typename 7−9
type-parameter 14−2
types
implementation-defined 17−5
implementation-defined exception 17−15
type-specifier
`bool` 7−8
`wchart` 7−8
type-specifier 7−6

U

`U` suffix 2−8
`u` suffix 2−8
`uflow`
`basic_filebuf` 27−62
`basic_streambuf` 27−29
unary
expression 5−13
minus operator 5−14
operator 5−13/14
operator, interpretation of 13−20
operator, overloaded 13−20
plus operator 5−14
unary-expression 5−13
`unary_function` 20−8
`unary_negate` 20−11
unary-operator 5−14
`uncaught\xception` 18−19
`#undef` 16−6, 17−10
undefined 17−2, 10/11, 13, 18−21, 21−16, 24−23, 26−14/19, 24, 28/29, 27−16
arithmetic exception 5−1
behavior 1−3
change to `const` object 7−7
change to string literal 2−11
`delete` 5−19
deleted object 3−27
division by zero 5−1, 22
escape sequence 2−10
floating point conversion 4−4
function call 5−11/12
left shift 5−23
member function call 9−6
modulus zero 5−1
overflow 5−1
pointer comparison 5−22, 24
pointer to function cast 5−11
pure virtual function call 10−11
string literal concatenation 2−11
unrepresentable integral value 4−4
`underflow`
`basic_filebuf` 27−62
`basic_streambuf` 27−29
`basic_stringbuf` 27−53
`strstreambuf` D−8
`underflow_error`
`underflow_error` 19−4
`underflow_error` 19−4
underlying
type, enumeration 7−11
type, `wchar_t` 3−32
underscore
character 17−11
in identifier _ 2−6
`unexpected` 18−19
`unexpected()` 15−9
`unexpected_handler` 17−12, 18−18

unget, basic_istream 27−40
uninitialized variable, indeterminate 8−16
uninitialized_copy 20−18
uninitialized_fill 20−18
uninitialized_fill_n 20−18
union 3−33, 9−10
 access control, anonymous 9−10
 anonymous 9−10
 at namespace scope, anonymous 9−10
 class versus 9−1
 constructor 9−10
 default member access 9−1
 destructor 9−10
 global anonymous 9−10
 initialization 8−20, 9−10
 member function 9−10
 restriction 9−10, 12−2
 restriction, anonymous 9−10
 type specifier 9−1
unique 25−16
 list 23−20
unique_copy 25−16
unit
 instantiation 2−2
 translation 17−9/10
unitbuf 27−21
universal-character-name 2−2
universal-character-name 2−2
unknown argument type 8−9
unnamed
 bit-field 9−11
 class 7−6
 object 12−2
 parameter example 8−15
unqualified name 3−9
unqualified-id 5−2
unrepresentable integral value, undefined 4−4
unsetf, ios_base 27−13
unshift, codecvt 22−18
unsigned
 arithmetic 3−32
 char type 3−32
 int type 3−32
 integer conversion, signed 4−4
 literal 2−8
 long type 3−32
 short type 3−32
 type 3−32
 type specifier 7−8
 typedef and 7−2
unsigned-suffix 2−8
unspecified 18−11/13, 15, 21−13, 25−19, 26−17, 27−53/D−5, D−7/8
 address of member function 17−14
 allocation 9−5, 11−3
 argument to constructor 5−18
 behavior 1−3
 constructor and new 5−18
 order of argument evaluation 5−6
 order of evaluation 3−23, 5−1
 order of evaluation new 5−18
 order of function call evaluation 5−6
 pointer comparison 5−24
unwinding, stack 15−4
up, name look 3−9
upper_bound 25−21
uppercase 17−6, 11
uppercase 27−21
usefacet, locale 22−9
user-defined
 conversion 12−4/6
 conversion and name hiding 12−6
 conversion, implicit 12−5
 conversion, inheritance of 12−7
 conversion, virtual 12−7
using-declaration 7−15
using-directive 7−20
usual arithmetic conversions 5−2
<utility> 20−5

V

va_end 17−11
<valarray> 26−9
valarray 26−12, 25
 apply 26−18
 cshift 26−17
 length 26−17
 max 26−17
 min 26−17
 operator! 26−16
 operator!= 26−19
 operator% 26−18
 operator%= 26−16
 operator& 26−18
 operator&& 26−18/19
 operator&= 26−16
 operator* 26−18
 operator*= 26−16
 operator+ 26−16, 18
 operator+= 26−16
 operator- 26−16, 18
 operator-= 26−16
 operator/ 26−18
 operator/= 26−16
 operator< 26−19
 operator<< 26−18
 operator<<= 26−16
 operator<= 26−19
 operator= 26−14/15
 operator== 26−19
 operator> 26−19
 operator>= 26−19
 operator>> 26−18
 operator>>= 26−16
 operator[] 26−15
 operator^ 26−18
 operator^= 26−16
 operator| 26−18
 operator|= 26−16
 operator|| 26−18/19
 operator~ 26−16
 resize 26−18
 shift 26−17
 sum 26−17
 valarray 26−14
 valarray 26−14
 ~valarray 26−14
~valarray, valarray 26−14
va_list 17−11
value
 call by 5−5
 null member pointer 4−5
 null pointer 4−4
 of char literal, implementation-defined 2−10
 of enumerator 7−10
 of multicharacter literal, implementation-defined 2−9
 representation 3−30
 undefined unrepresentable integral 4−4
variable
 argument list 8−9

indeterminate uninitialized 8−16
parameter list 5−5, 8−9
parameter list example 8−9
<vector> 23−11
vector 23−24
assign 23−26
capacity 23−26
erase 23−27
insert 23−27
operator< 23−26
operator== 23−26
reserve 23−26
resize 23−27
vector 23−26
vector 23−26
vector<bool> 23−27
operator< 23−29
operator== 23−29
vertical tab 2−9
viable function 13−5
virtual
base class 10−2
base class DAG 10−3
base class dominance 10−5
base class initialization 12−15, 21
base class initialization, order of 12−14
destructor 12−8
destructor, pure 12−8
function 10−6
function access 11−9
function call 10−10
function call, constructor and 12−17
function call, destructor and 12−17
function call, undefined pure 10−11
function definition 10−8
function definition, pure 10−10
function example 10−8
function example, pure 10−10
function, pure 10−10/11
member function 17−11, 14
user-defined conversion 12−7
virtual
and friend 10−8
and multiple inheritance 10−8
specifier 7−5
visibility 3−9
void
parameter 8−9
type 3−33
type specifier 7−8
void& 8−6
void*
conversion, pointer to member 4−5
pointer comparison 5−24
type 3−33
volatile 3−34
constructor and 9−8, 12−1
destructor and 9−8, 12−7
implementation-defined 7−8
member function 9−7
overloading and 13−2
type 7−6
type specifier 7−8

W

wcerr 27−6
<wchar.h> D−1
wchart type-specifier 7−8
wchar_t 2−9, 11, 17−7, 21−31
implementation-defined 3−32
type 3−32
underlying type 3−32
wcin 27−6
wclog 27−6
wcout 27−6
wcschr 21−32
wcspbrk 21−32
wcsrchr 21−32
wcsstr 21−32
<wctype.h> D−1
well-formed program 1−4
wfilebuf 27−4, 59
wfstream 27−4
what
bad_alloc 18−14
bad_cast 18−16
bad_exception 18−18
bad_typeid 18−16
exception 18−17
ios_base::failure 27−10
while statement 6−3/4
white
space 2−3
space 2−4
wide string literal 2−11
wide-character 2−9
string, null-terminated 17−7
widen
basic_ios 27−18
ctype 22−12
ctype<char> 22−16
width, ios_base 22−10, 27−13
wifstream 27−4, 59
wios 27−7
wistream 27−4, 31
wistringstream 27−4, 51
wmemchr 21−32
wofstream 27−4, 59
wostream 27−4, 31
wostringstream 27−4, 51
write, basic_ostream 27−47
ws 27−36, 41
wstreambuf 27−4, 22
wstreamoff 27−7
wstringbuf 27−4, 51
wstringstream 27−4

X

xalloc, ios_base 27−14
xsgetn, basic_streambuf 27−28
xsputn, basic_streambuf 27−30
X(X&) —see copy constructor , 19

Z

zero
pointer 4−4
undefined division by 5−1, 22
undefined modulus 5−1
width of bit-field 9−11
zero-initialization 8−16