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
test 11–1
test, anonymous union 9–10
test default 11–1
test, member function and 12–1
test, overloading resolution and 10–5
declaration 11–4
equation, member name 11–4
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 26–28
acos 26–19, 31
addition operator 5–21
additive operator 5–21
additive expression 5–21
address
of bit-field 9–11
of bit-field restriction 9–11
of constructor 12–2
of cv-qualified name 5–14
of specified function, unspecified 17–13
of overloading function 5–14, 13–18
address-of operator 5–13
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 3–30
allocation
function 3–25, 5–17, 12–9
implementation defined bit-field 9–11
new, storage 5–15
unspecified 9–5, 11–3
Allocator requirements 20–2
allocate 20–15
allowing an exception 15–7
alternate definition 17–11
always_nocvt, codecvt 22–17
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
anachronism C–11
C function definition C–12
assignment to this C–12
cast of pointer to member C–13
free store and constructor C–12
free store and destructor C–12
memory management C–12
nonnested class C–13
old style base class initializer C–12
old style function definition C–12
overload keyword C–11
pointer to member conversion C–13
scope of nested class C–13
this and constructor C–12
this and destructor C–12
AND
operator, bitwise 5–24
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, extension to C C–1
union, global 9–10
union restriction 9–10
any, bitset 23–45
append, basic_string 21–16
apply, valarray 26–17
arg, complex 26–8
argc 3–21
argument 1–2, 17–12/13, 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
deduction, template 14–36
evaluation of default 8–12/13
evaluation of 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 14–35
substitution 16–5
template 14–5
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[] 3–21
arithmetic
conversions, usual 5–2
exception 5–1
exception, undefined 5–1
extension to C single precision C–1
pointer 5–22
type 3–32
unsigned 3–32
array
bound 8–8
const 3–33
declaration 8–8
declarator () 8–8
declarator, multidimensional 8–8
delete 5–18
element 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 5–17
of class objects initialization 8–20, 12–12
order of execution, constructor and 12–11
order of execution, destructor and 12–7
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 3–32, 8–10
array-to-pointer conversion 4–2
arrow operator —see class member access operator
as-if rule 1–5
asin 26–19, 31
asm
declaration 7–23
implementation-defined 7–23
assembler 7–23
<assert.h> 17–9/D–1
assign
basic_string 21–17
deqe 23–13
list 23–17
vector 23–24
Assign: able requirements 23–1
assignment
and initialization, overloaded 12–12
and value 5–26
conversion by 5–26
expression 5–26
extension to C memberwise C–2
operator 5–26, 17–7
operator, copy 12–20
operator, overloaded 13–21
operator restriction, copy 12–21
reference 8–21
to class object 5–26
to reference 5–26
to this anachronism C–12
assignment-expression 5–26
assignment-operator 5–26
at.basic_string 21–16
atan 26–19, 31
atan2 26–19, 31
atexit 3–23, 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 3–24
automatic initialization 6–6/7
auto_ptr 20–18
auto_ptr 20–19
auto_ptr 20–19
get 20–19
operator* 20–19
operator-> 20–19
operator= 20–19
release 20–19
~auto_ptr 20–19
~auto_ptr, auto_ptr 20–19

B
back_inserter 24–16
back_insert_iterator 24–15
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 27–18
bad_alloc 5–18, 18–10, 14
bad_alloc 18–13
bad_alloc 18–13
bad_alloc 18–13
bad_cast::what, implementation-defined 18–13
bad_cast 5–8, 18–15
bad_cast 18–15
bad_cast 18–15
operator= 18–15
what 18–15
bad_cast::what, implementation-defined 18–15
bad_exception 18–17
bad_exception 18–17
bad_exception 18–17
operator= 18–17
what 18–17
bad_exception::what, implementation-defined 18–17
bad_typeid 5–9, 18–15
bad_typeid 18–15/16
bad_typeid 18–15/16
operator= 18–16
what 18–16
bad_typeid::what, implementation-defined 18–16
base
class 17–11, 14
class 10–12
class access 11–3
class cast 5–11
class constructor order of execution 12–2
class destructor order of execution 12–7
class, direct 10–1
class, indirect 10–1
class initialization 12–12/13
class initialization, order of 12–14
class initializer 8–15
class initializer anachronism, old style C–12
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, 56
basic_filebuf 27–57
basic_filebuf 27–57

read 27–37
readsome 27–38
seekg 27–38
sentry 27–32
sync 27–38
tellg 27–38
unget 27–38
~sentry 27–33
basic_istreambuf_iterator 27–4
basic_istream<char> 27–29
basic_istream<wchar_t> 27–29
basic_istreamstream 27–4, 52
basic_istreamstream<char> 27–52
basic_istreamstream<wchar_t> 27–52
basic_ofstream 27–4, 62
basic_ofstream<char> 27–63
basic_ofstream<wchar_t> 27–63
close 27–63
is_open 27–63
open 27–63
rdbuf 27–63
basic_ifstream 27–64
basic_ifstream<char> 27–64
basic_ifstream<wchar_t> 27–64
basic_ifstreamstream 27–64
basic_ifstreamstream<char> 27–64
basic_ifstreamstream<wchar_t> 27–64
basic_ios 27–4, 15
bad 27–18
basic_ios<char> 27–7
basic_ios::failure 27–18
argument, implementation-defined
basic_iosstream 27–39
basic_iosstream<char> 27–39
basic_iosstream<wchar_t> 27–39
basic_streambuf 27–39
basic_streambuf<char> 27–39
basic_streambuf<wchar_t> 27–39
basic_ostream 27–4, 30
basic_ostream<char> 27–4, 30
basic_ostream<wchar_t> 27–4, 30
basic_ostreamstream 27–4, 52
basic_ostreamstream<char> 27–52
basic_ostreamstream<wchar_t> 27–52
gcount 27–35
get 27–35
getline 27–36/37
ignore 27–37
open 27–37
operator bool() 27–33
peek 27–37
putback 27–38
pubseekoff 27–24
pubseekpos 27–24
pubsetbuf 27–24
C

anonymous union, extension to C–1
class, extension to C–1
const, extension to C–1
dangerous extension to C–1
declaration statement, extension to C–1
delete, extension to C–1
destructor, extension to C–2
direct call, virtual function call
expression evaluation, difference from C–1
extension to C–1/2
function definition anachronism C–12
header 17–9, 11, 17–13/D–1
implementation-defined extension to C–11
inline function, extension to C–1
library, Standard 17–1, 6, 17–8/3/C–13, C–15
linkage to 7–24
memberwise assignment, extension to C–2
memberwise initialization, extension to C–2
multiple inheritance, extension to C–2
new, extension to C–1
overloading delete, extension to C–2
overloading, extension to C–1
overloading new, extension to C–2
pointer to member, extension to C–2
protected, extension to C–2
reference type, extension to C–1
single precision arithmetic, extension to C–1
summary, compatibility with C–1
summary, compatibility with ISO C–2
type checking, extension to C–1
user-defined type, extension to C–1
void* pointer type extension to C–1
volatile, extension to C–2
call —see also function call, member function call, overloaded function call, virtual function call
by reference 5–5
by value 5–5
operator function 13–20
pseudo destructor 5–6
calloc 20–20/C–16
candidate functions 14–26
capacity
basic_string 21–15
type 3
basic ostream operator 27–33
type 3
bool alpha 27–19
Boolean
bit-fields 9–11
conversion 4–5
literal 2–11
type 3–31
type 3–32
boolean-literal 2–11
bound array 8–8
bound, of array 8–8
bound pointer to member function, undefined C–13
break statement 6–56
built-in type —see fundamental type byte 5–15
string, null-terminated 17–6
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 3–31
string 2–11
type 3–31
underscore 17–10/11
cchar_traits
eos 21–21
eq 21–21/23
length 21–13/14, 16/19, 21/23, 25/26
checking
point of error 14–17
syntax 14–17
cin 27–5
<cmath> C–15
class 3–33, 9–1
abstract 10–10
access and friend 11–6
anachronism, nonnested C–13
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
extension to C C–1
gslice 26–22
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 3–24
member storage duration 3–26
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 5–26
object, const 3–33
object copy 12–19
object copy — see also copy constructor
object initialization 8–18, 12–11
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 5–17
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 23–42
template partial specializations 14–12
template specialization 14–5
unnamed 7–6
class
type specifier 9–1
versus struct 9–1
versus union 9–1
class, locale 22–9
class_table, ctype<char> 22–15
class-key 9–1
class-name 9–1
class-specifier 9–1
clear
basic_ios 27–18
basic_string 21–15
<limits> 18–8, 23–43/D–5
<locale> 17–6, 22–45/C–16
clog 27–5
close
basic_filebuf 27–58, 64
basic_ifstream 27–62
basic_ofstream 27–63
messages 22–41
<cmath> 26–30
codecvt 22–16
always_noconv 22–17
do_always_noconv 22–19
do_encoding 22–19
do_in 22–18
do_length 22–19
do_max_length 22–19
do_out 22–18
do_unshift 22–18
encoding 22–17
in 22–17
length 22–17
max_length 22–17
out 22–17
unshift 22–17
codecvt_varname 22–19
collate 22–30
compare 22–31
do_compare 22–31
do_hash 22–31
do_transform 22–31
hash 22–31
transform 22–31
collate_varname 22–31
comma
operator 5–27
operator, side effects and 5–27
comment 2–3
/** */ 2–4
// 2–4
compare
basic_string 21–24
collate 22–31
comparison
  function 17–1
  pointer 5–23
  pointer to function 5–23
  undefined pointer 5–22/23
  unspecified pointer 5–23
void* pointer 5–23
compatibility
  with C summary C–1
  with ISO C summary C–2
compilation, separate 2–1
compiler control line — see preprocessing directive
complete object 1–4
completely defined object type 9–4
<complex> 26–2
  complex 26–3
  abs 26–8
  arg 26–8
  complex 26–5
  conj 26–8
  cos 26–8
  cosh 26–8
  exp 26–8
  imag 26–8
  log 26–8
  log10 26–8
  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
  polar 26–8
  pow 26–8
  real 26–8
  sin 26–8
  sinh 26–9
  sqrt 26–9
  tan 26–9
  tanh 26–9
  component 17–1
compound
  statement 6–1
  type 3–32
<compound-statement> 6–1
concatenation
  string 2–11
  undefined string literal 2–11
condition 6–2
  conditional
    expression operator 5–25
    inclusion 16–2
  conditional-expression, throw-expression in 5–25
  conditions, rules for 6–2
  conj, complex 26–8
consistency
  example, linkage 7–3
  linkage 7–3
  linkage specification 7–25
  type declaration 3–21
const
  cast 5–12
  member initialization 12–14
*const example 8–5
const 3–33
  array 3–33
  class object 3–33
  constructor and 9–8, 12–1
  destructor and 9–8, 12–7
  example 8–5
  extension to C C–1
  initialization 7–7, 8–18
  linkage of 3–19, 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 5–27
  expression, pointer to member 5–14
  initializer 9–4
  null pointer 4–4/5
  pointer declaration 8–5
  pointer example 8–5
  constant-expression 5–27
  constant-initializer 9–4
  constructor 12–1
  address of 12–2
  anachronism, free store and C–12
  anachronism, this and C–12
  and abstract class 10–11
  and array order of execution 12–11
  and const 9–8, 12–1
  and initialization 12–11
  and initialization example 12–11
  and member function 12–2
  and new 5–17
  and new, unspecified 5–18
  and return 6–6
  and static objects order of execution 12–12
  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 12–4
  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 15–3
  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
continue
  in for statement 6–5
  statement 6–5/6
control line — see preprocessing directive
conversion 17–4
conversion
  Boolean 4–5
  ambiguity, class 10–6
  anachronism, pointer to member C–13
  and name hiding, user-defined 12–6
argument 8–9
array-pointer 4–2
array-to-pointer 4–2
by-assignment 5–26
by constructor 12–4
class 12–4
derived-to-base 13–14
explicit-type 8–see-casting
floating-point 4–4
floating-integral 4–4
function 8–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 12–6
inheritance-of-user-defined 12–6
integer 4–4
i-value-to-rvalue 4–2
operator 12–5
overload-resolution-and-pointer 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 5–10
to-enumeration-type, static_cast, 5–10
to rvalue, i-value 4–2
type 12–5
undefined-floating-point 4–4
user-defined 4–4/5
virtual-user-defined 12–6
conversion-function-id 12–5
conversions
qualification 4–2
usual-arithmetic 5–2
copy
assignment-operator 12–20
assignment-operator 12–19
assignment-operator, implicitly-declared 12–21
assignment-operator-restriction 12–21
class object 12–19
class-parameter 12–19, 17–7
constructor, implicitly-declared 12–19
constructor-restriction 12–20
copy 25–12
basic_string 21–20
copy_backward 25–13
CopyConstructible requirements 20–2
copyfmt, basic_ios 27–17
cos 26–19, 31
complex 26–8
cosh 26–19, 31
complex 26–8
count 25–11
bitset 23–45
count_if 25–11
cout 27–5
cplusplus 16–9
<cstdio> 17–10, 18–19
cshft, valarray 26–17
<csignal> 18–19
<cstdlib> 8–10, 17–10, 18–19
<cstring> 5–15, 22, 18–1/C–16
<ctime> 27–56, 56, 58, 60, 27–64/C–16
<ctime> 3–22, 24, 17–9, 18–9, 19, 20–20, 21–29,
25–27, 26–30/C–16
c_str, basic_string 21–20
cstring 17–6, 20–20, 21–29/D–5, D–10/C–16
<ctime> 18–19, 20–20, 22–2/C–16
c_type 12–12
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 22–12
is 22–11
narrow 22–11
scan_is 22–11
scan_not 22–11
tolower 22–11
toupper 22–11
widen 22–11
c_type_byname 22–13
c_type_byname<char> 22–16
c_type<char>
classic_table 22–15
c_type<char> 22–14
c_type<char> 22–14
is 22–14
narrow 22–15
scan_is 22–15
scan_not 22–15
tolower 22–15
toupper 22–15
widen 22–15
~c_type<char> 22–14
~c_type<char>, c_type<char> 22–14
<c_type.h> D–1
cv-qualified-name, address-of 5–14
cv-qualified 3–33
cv-qualifier 8–2
<cwchar> 17–6, 11, 21–29/C–16
<cwctype> 17–11, 21–28

D

DAG
multiple-inheritance 10–3
nonvirtual-base-class 10–3
virtual-base-class 10–3
dangerous-extension-to-C 11–11
data
member 8–see-member
member.static 9–8
data, basic_string 21–21
date_order, time_get 22–33
deallocating 8–see-delete
function 3–26, 5–19, 12–9
dec 27–20, 44
decimal-literal 2–8
decimal-literal 2–7
decimal-point-character 17–6
decimal-point, numpunct 22–29
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 3–21
constant pointer 8–5
default argument 8–11
definition versus 8–1
eellipsis in function 5–5, 8–9
enumerator point of 3–5
element 3–2, 8–11
example, function 8–10
eextern 3–1
eextern 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 3–21
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
statement, extension to C 3–1
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 12–3
meaning of 8–4
multidimensional array 8–8
declarator 8–1
declarator-id 8–2
decl-specifier 7–2
decrement
operator 5–7, 13, 15
operator, overloaded 13–22
deduction, template argument 14–36
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 17–1, 4
constructor 12–2
constructor and initialization 12–11
constructor and new 5–17
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
declaration 3–1, 17–1
alternate 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
eempty class 9–1
enumerator 3–2
enumerator point of 7–11
example 3–1
eexample, function 8–15
eexample, nested class 9–12, 11–10
efunction 3–2
efunction 8–14
elocal class 9–13
emember function 9–5
e 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
t virtual function 10–8
[]], class 9–1
definitions, implementation-generated 3–2
delete
array 5–18
object 5–18
delete 3–24, 5–18/19, 12–9
destructor and 5–19, 12–8
eexample 12–10
eexample, destructor and 12–10
eexample, scope of 12–10
extension to C 3–1
extension to C overloading 3–2
operator 17–11, 18–11, 20–20
overloading and 3–26
type of 12–10
undefined 5–19
delete[], operator 17–11, 18–12
deleted object, undefined 3–26
delete-expression 5–18
typedef
entity 3
±
ends
27
±
endl
27
±
equilibrating
—see variable declaration
cast 5–8, 18–15
initialization 3–22
storage duration 3–24, 5–16
type 1–2
E
E suffix 2–10
eback, basic_streambuf 27–25
effect, side 1–5
egp_ptr, basic_streambuf 27–25
elaborated
class name 7–9, 9–2/3
enum name 7–9
type specifier — see elaborated class name
elaborated-type-specifier 7–9
#else 16–2
elimination of temporary 12–2
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 2–11
encoding, codecvt 22–17
end, basic_string 21–15
#endif 16–3
endl 27–43, 46
end-of-file 23–46
ends 27–46
entity 3–1
enum name, typedef 7–6
enum 3–33
declaration () 7–10
name, elaborated 7–9
overloading and 13–2
type of 7–10/11
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 5–10
type, static_cast, conversion to 5–10
underlying type 7–11
everifier
class, scope of 7–11
definition 3–2
member 7–11
point of declaration 3–5
point of definition 7–11
redefinition 7–11
restriction 7–11
value of 7–10
enumerator 7–10
environment, program 3–22
eof, basic_ios 27–18
eos, char_traits 21–21
egp_ptr, basic_streambuf 27–25
eq, char_traits 21–21/23
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 14–7
type 7–5, 9–2
equivalent
parameter declarations 13–2
parameter declarations, overloading and 13–2
erase
basic_string 21–18
deque 23–14
list 23–18
vector 23–25
<errno.h> D–1
error
checking, point of 14–17
directive 16–8
#error 16–8
escape
character — see backslash
sequence 2–9
sequence, undefined 2–9
escape-sequence 2–8
evaluation
difference from C expression C–1
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
class pointer 8–5
class constructor 12–2
class constructor and initialization 12–11
declaration 3–2, 8–11
declarator 8–2
definition 3–1
delete 12–10
derived class 10–1
destructor and delete 12–10
destructor 10–1
destination 8–9
equater 7–11
explicit destructor call 12–8
explicit qualification 10–5
friend 9–2
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 11–4
named class 9–12
nested class 9–12
nested class definition 9–12, 11–10
nested class forward declaration 9–12
nested type name 9–13
basic_string 21–21
find_end 25–10
find_first_not_of.basic_string 21–23
find_first_of 25–10
basic_string 21–22
find_if 25–10
find_last_not_of.basic_string 21–23
find_last_of 21–22
float Digit 2–7
fixed 27–20
flags.ios_base 22–10, 27–12
flip.bitset 23–44
float
literal 2–10
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 2–10
point literal, type of 2–10
point promotion 4–4
point type 3–31
point type 3–32
point type, implementation-defined 3–32
floating-integral conversion 4–4
floating-literal 2–10
floating-suffix 2–10
float_round_style 18–7
floor 26–31
flush 27–12, 32, 42, 46
basic_ostringstream 27–45
fmtflags
ios 27–46
ios_base 27–10
fopen 27–58
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 also parameter
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, 14/15
fpos 27–14
fpos 27–14
state 27–14
fractional-constant 2–10
free
store —see also new, delete
store and destructor anachronism C–12
store and destructor anachronism C–12
free 20–20
freestanding implementation 17–8
freeze
ostrstream D–10
strstreambuf D–6
frexp 26–31
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
element example 9–2
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 14–11
virtual and 10–8
front_insert_iterator 24–17
front_insert_iterator 24–16
front_insert_iterator 24–17
operator* 24–17
operator++ 24–17
operator= 24–17
fseek 27–58
<fstream> 27–55
fstream 27–4
full-expression 1–6
function
—see also friend function, member function, inline
function, virtual function
allocation 3–25, 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 5–4, 13–19
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 5–23
conversion, pointer to 4–2
deallocation 3–26, 5–19, 12–9
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 1) 8–9
definition 8–14
definition 3–2
definition anachronism, C
definition anachronism, old style
definition example 8–15
global 17–10, 13
handler 17–1
linkage specification overloaded 7–25
modifier 17–1
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
pointer to member 5–21
fundamental function — see return
return — see return type
scope 3–6
specifier 7–4
template 14–34
template overload resolution 14–41
type 3–32, 8–W/10
typedef 8–10
virtual 13–5
virtual — see virtual function
virtual member 17–11, 13
<functional> 20–7
function-body 8–14
function-definition 8–14
function-like macro 16–4
functions, candidate 14–26
function-specifier 7–4
function-to-pointer conversion 4–2
function-try-block 15–1

fundamental


G

gDispose

g ACPI

g_count

gCount

generate

generate

get

generate

generate

generate

get

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

generate

h

handle

handler

exception 15–4, 17–14
function 17–1
incomplete type in exception 15–4
handler 15–1
handler-seq 15–1
has facet, locale 22–9
hash, collate 22–31
header, C 17–9, 11, 17–13/D–1
header-name 2–5
headers, C++ 17–8
hex number 2–10
hex 27–20
hexadecimal literal 2–8
hexadecimal-digit 2–7
hexadecimal-escape-sequence 2–9
hexadecimal-literal 2–7
hex-quad 2–2
hiding — see name hiding
horizontal tab 2–9
hosted implementation 17–8

I

id, qualified 5–3
id, locale 22–7
identifier 2–6, 5–3, 7–1
_, underscore in 2–6
identifier 2–5
identities and overloading, operator 13–20
id-expression 5–3
id-expression 5–3
#if 16–2, 17–13
if statement 6–23
#ifdef 16–3
#ifndef 16–3
ifstream 27–4, 56
ignore, basic_istream 27–37
ill-formed program 1–3
imag 26–6
complex 26–8
imbue

basic_filebuf 27–61
basic_ios 27–17
basic_streambuf 27–26
implicit implementation-defined types 4
± manip ± implementation-dependent 27–13
± implementation-defined 17
± default constructor and 12
± default constructor — see default constructor
implementation-defined alignment of bit-field 9–11
± defined bit-field allocation 9–11
± defined division 5–21
± defined modulus 5–21
± defined pointer integer conversion 5–11
± defined pointer subtraction 5–22
± defined right shift 5–23
± defined sizeof expression 5–15
± defined type of ptrdiff_t 5–22
± defined type of size_t 5–15
freestanding 17–8
hosted 17–8
limits 1–3
implementation-defined 4–4, 8–22, 17–8, 11, 18–1, 10, 13, 15/17, 27–13, 52, 27–60/C–16
± _STDC__ 16–9
alignment requirement 3–30
± asm 7–23
± bad_alloc::what 18–13
± bad_cast::what 18–15
± bad_exception::what 18–17
± bad_typeid::what 18–16
± basic_ios::failure argument 27–18
± behavior 1–3
behavior reentrancy 17–14
exception specifications 17–14
exception types 17–14
exception::what message 18–17
± extension to C C–11
± filebuf 27–61
± floating point conversion 4–4
± floating point type 3–32
± generation of temporary 12–2
linkage of main() 3–22
linkage specification 7–24
± object linkage 7–26
order of evaluation of function arguments 8–13
parameters to main() 3–21
± sign of bit-field 9–11
± sign of char 3–31
± sizeof integral type 3–31
± sizeof type 3–31
± 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
± value of char literal 2–10
± value of multicharacter literal 2–9
± volatile 7–8
± wchar_t 3–32
implementation-dependent 27–32, 42
implementation-generated definitions 3–2
implementation-specified 27–46
implementation-defined types 17–5
± implicit conversion 4–1, 12–4
± conversion sequence 13–13
± conversion sequences implied object parameter 13–5
destructor call 12–8
± instantiation, template 14–27
± object argument 13–5
± user-defined conversion 12–6
± implicitly-declared copy assignment operator 12–21
copy constructor 12–19
default constructor 12–2
default constructor — see default constructor
implied object parameter 13–5
± object parameter, implicit conversion sequences 13–5
± in_codecv 22–17
± in_avail.basic_streambuf 27–24
± includes 25–22
± inclusion conditional 16–2
source file 16–3
± incomplete class, cast to 5–20
± type 3–2/3, 5, 30, 4–2, 5–4/6, 8/9, 14/15, 19, 21, 26, 10–1
type, example of 3–30
type in exception handler 15–4
± increment bool 5–7, 15
operator 5–7, 13, 15
operator, overloaded 13–22
indeterminate uninitialized variable 8–16
indirect base class 10–1
± indirect_array 26–26
± fill 26–27
± indirect_array 26–27
± indirect_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
± indirect = 5–14
operator 5–13
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 12–6
± Init.ios_base::Init 27–12
± init.ios_base 27–32, 42
± Init.ios_base::Init 27–12
± init-declarator 8–1
± init-declarator-list 8–1
± initialization 8–15
and goto 6–6
and new 5–17
array 8–18
± array of class objects 8–20, 12–12
± auto 6–7
± auto object 8–15
± automatic 6–67
± base class 12–12/13
± character array 8–20
± class member 8–16
± class object 8–18, 12–11
± class object — see also constructor
± const 7–7, 8–18
± const member 12–14
± constructor and 12–11
default 8–16
± default constructor and 12–11
definition and 7–2
dynamic 3–22
± example, constructor and 12–11
extension to C memberwise C–2
in block 6–6
jump past 6–3, 6
local static 6–7
member 12–12
member object 12–13
order of 3–22, 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 3–22
static member 9–9
static object 3–22, 8–15/16
struct 8–18
union 8–20, 9–10
virtual base class 12–14, 20
initializer 8–15
base class 8–15
constant 9–4
list () 8–18
member 8–15
scope of member 12–15
temporary and declarator 12–3
initializer 8–15
initializer-clause 8–15
initializer-list 8–15
inline 17–13
friend function 11–7
function 7–4
function, extension to C C–1
member function 9–5
inline
linkage of 3–19
specifier 7–4
inner_product 26–28
inplace_merge 25–21
input_iterator_tag 24–9/10
insert
basic_string 21–17
deque 23–14
list 23–18
vector 23–25
inserter 24–18
insert_iterator 24–17
insert_iterator 24–18
insert_iterator 24–18
operator* 24–18
operator++ 24–18
operator= 24–18
instantiation
explicit 14–30
point of 14–25
template implicit 14–27
unit 2–2
int, bool promotion to 4–3
int
type 3–31
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 3–31
type 3–32
type, implementation-defined sizeof 3–31
value, undefined unrepresentable 4–4
internal linkage 3–19
internal 27–20
interpretation
of binary operator 13–20
of unary operator 13–20
invalid_argument 19–2, 23–42/43
invalid_argument 19–2
invalid_argument 19–2
invocation, macro 16–5
<iomanip> 27–29
<iostream> 27–6
ios 27–4, 7
fmtflags 27–46
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 27–10
iword 27–13
openmode 27–11
precision 22–10, 27–12
pword 27–13
register_callback 27–14
seekdir 27–11
setf 27–12
sync_with_stdio 27–13
unsetf 27–12
width 22–10, 27–13
xalloc 27–13
ios_base::failure 27–9
failure 27–9
what 27–9
ios_base::Init 27–11
Init 27–12
~Init 27–12
<iiosfwd> 27–2
iostream 27–10
<iostream> 27–5
is
cstype 22–11
cstype<char> 22–14
isalnum 22–9
isalpha 22–9
iscntrl 22–9
isdigit 22–9
isgraph 22–9
islower 22–9
ISO C summary, compatibility with C–2
<i64.h> D–1/C–15
is_open
basic_filebuf 27–57, 64
basic_ifstream 27–62
basic_ofstream 27–63
isprint 22–9
ispunct 22–9
isspace 22–9
<iiostream> 27–29
iostream 27–4, 29
operator>> 27–33
istreambuf_iterator 24–21
equal 24–24
istreambuf_iterator 24–23
istreambuf_iterator 24–23
operator!= 24–24
operator* 24–23
operator++ 24–23
operator== 24–24
proxy 24–22
istream_iterator 24–19
operator== 24–20
istringstream 27–4, 48
iostreamstream 27–4
istreamstream 27–4
istreamstream 27–4
rddbuf 27–4
str 27–4
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 27–13

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

K
keyword A–1
anachronism, overload C–11
list 2–6

L
prefix 2–9, 11
suffix 2–8, 10
l suffix 2–8, 10
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 5–22
shift, undefined 5–22
left 27–20
length of name 2–6
length
char_traits 21–13/14, 16/19, 21/23, 25/26
codecvt 22–17
valarray 26–16
length_error 19–2, 21–9
length_error 19–2
length_error 19–2
less
than operator 5–23
than or equal to operator 5–23
less 20–10
less_equal 20–10
LessThanComparable requirements 20–2
lexical conventions 2–1
lexicographical_compare 25–26
library
C++ Standard 17–1, 11/12, 14
Standard C 17–1, 6, 17–8C–13, C–15
limits, implementation 1–3
<windows.h> 18–2
<limits.h> D–1
#line 16–8
linkage 3–1, 19
consistency 7–3
consistency example 7–3
external 3–19, 17–9/11
implementation-defined object 7–26
internal 3–19
of class 3–20
of const 3–19, 7–3
of enumeration 3–20
of extern 7–3
of friend function 11–6
of inline 3–19
of main(), implementation-defined 3–22
of static 3–19, 7–3
specification 7–24
specification class 7–24
specification consistency 7–25
specification, extern 7–24
specification, implementation-defined 7–24
specification object 7–26
specification overloaded function 7–25
to C 7–24
linkage-specification 7–24
list
keyword 2–6
operator 2–7, 13–19
{ }, initializer 8–18
<li> 23–9
list 23–15
assign 23–17
erase 23–18
insert 23–18
merge 23–19
remove 23–18
resize 23–17
reverse 23–19
sort 23–19
splice 23–18
unique 23–19
literal 2–7, 5–2
base of integer 2–8
character 2–9
decimal 2–8
double 2–10
float 2–10
floating point 2–10
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
locale
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 26–19, 31
complex 26–8
log10 26–19, 31
complex 26–8
logical
AND operator 5–25
AND operator, side effects and 5–25
OR operator 5–25
OR operator, side effects and 5–25
negation operator 5–13/14
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 2–10
double type 3–32
literal 2–8
type 3–31
type specifier 7–8
type, unsigned 3–32
typedef and 7–2
longjmp 18–20
long-suffix 2–8
look up, name 3–8
lookup
argument-dependent 3–12
member name 10–4
name 3–1
template name 14–16
lower_bound 25–20
lowercase 17–6
lvalue 3–34
assignment and 5–26
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() 3–21
implementation-defined linkage of 3–22
implementation-defined parameters to 3–21
parameters to 3–22
return from 3–22/23
make_heap 25–25
make_pair 20–6
malloc 20–20/C–16
management anachronism, memory C–12
<map> 23–28
map 23–29
operator< 23–32
operator== 23–32
operator[] 23–32
mask_array 26–25
fill 26–26
mask_array 26–25
mask_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^= 26–26
operator&= 26–26
operator| 26–26
masking macro 17–13
<math.h> D–1
max 25–25
valarray 26–16
max_element 25–25
max_length, codecvt 22–17
max_size, basic_string 21–15
meaning of declarator 8–4
member
---see also base class member
access operator, overloaded 13–22
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 14–20, 25
elaborated enum 7–9
global 3–7
hiding 3–5, 8, 5–2/3, 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 3–8
lookup 3–1
lookup, member 10–4
lookup, template 14–16
macro 16–5
overloaded function 13–1
overloaded member 9–4
point of declaration 3–5
qualified 3–13
reserved 17–10
resolution, template 14–16
scope of 3–4
space, label 6–1
unqualified 3–9
name locale 22–8
type_info 18–15
namespaces 7–12
namespace 17–7, 17–10/D–1
definition 7–12
scope 3–6
scope, anonymous union at 9–10
scope, global 3–7
std 17–10
namespaces 7–12
new-placement 5–15
new-type-id 5–15
next_permutation 25–26
noboolalpha 27–19
nondigit 2–5
non-nested class anachronism C–13
non-trivial constructor 12–2
destructor 12–7
nonvirtual base class DAG 10–3
nonzero-digit 2–7
norm, complex 26–8
noshowbase 27–19
noshowpoint 27–19
noshowpos 27–19
noti 20–11
not2 20–11
notation, syntax 1–3
not_equal_to 20–9
nounitbuf 27–20
nounppercase 27–20
NTBS 17–6, 27–58/D–10
static 17–6
nth_element 25–19
NTMBS 17–6
static 17–6
NTWCS 17–6/7
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 18–1
null-terminated
byte string 17–6

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 14–20, 25
elaborated enum 7–9
global 3–7
hiding 3–5, 8, 5–2/3, 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 3–8
lookup 3–1
lookup, member 10–4
lookup, template 14–16
macro 16–5
overloaded function 13–1
overloaded member 9–4
point of declaration 3–5
qualified 3–13
reserved 17–10
resolution, template 14–16
scope of 3–4
space, label 6–1
unqualified 3–9
name locale 22–8
type_info 18–15
namespaces 7–12
namespace 17–7, 17–10/D–1
definition 7–12
scope 3–6
scope, anonymous union at 9–10
scope, global 3–7
std 17–10
namespaces 7–12
new-placement 5–15
new-type-id 5–15
next_permutation 25–26
noboolalpha 27–19
nondigit 2–5
non-nested class anachronism C–13
non-trivial constructor 12–2
destructor 12–7
nonvirtual base class DAG 10–3
nonzero-digit 2–7
norm, complex 26–8
noshowbase 27–19
noshowpoint 27–19
noshowpos 27–19
noti 20–11
not2 20–11
notation, syntax 1–3
not_equal_to 20–9
nounitbuf 27–20
nounppercase 27–20
NTBS 17–6, 27–58/D–10
static 17–6
nth_element 25–19
NTMBS 17–6
static 17–6
NTWCS 17–6/7
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 18–1
null-terminated
byte string 17–6

DRAFT: 2 December 1996 Index 21
multibyte string 17–6
wide-character string 17–6
number
hex 2–10
octal 2–10
numeric type requirements 26–1
<numeric> 26–27
numeric_limits 3–32, 18–2
num_get 22–20
do_get 22–22
get 22–21
numpunct 22–28
decimal_point 22–29
do_decimal_point 22–29
do_grouping 22–29
do_thousands_sep 22–29
do_true_name do_falsename 22–30
grouping 22–29
thousands_sep 22–29	ruename fasename 22–29
numpunct_byname 22–30
num_put 22–24
do_put 22–25
put 22–25

O
object 1–4, 3–1, 34
class —see also class object
complete 1–4
definition 3–2
delete 5–18
destructor and placement of 12–8
destructor static 3–23
initialization, auto 8–15
initialization, static 3–22, 8–15/16
layout, access specifier and 11–3
lifetime 3–26
linkage, implementation-defined 7–26
linkage specification 7–26
representation 3–30
state 17–2
static local 3–24
storage duration, local 3–24
temporary —see temporary
type 1–4
type 3–31
type, completely defined 9–4
undefined deleted 3–26
unnamed 12–2
object-expression 5–1
object-like macro 16–4
observer function 17–2
oct 27–20
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 18–2C–16
ofstream 27–4, 56
old
style base class initializer anachronism C–12
style function definition anachronism C–12
one-definition rule 3–2
one’s complement operator 5–13/14
open
basic_filebuf 27–57, 64
basic_ifstream 27–62
basic_ofstream 27–63
messages 22–41
openmode, ios_base 27–11
operations on class object 9–1
operator
%= 5–26
-= 5–26
+= 5–15, 26
-= 5–26
/= 5–26
<<= 5–26
>>= 5–26
^= 5–26
additive 5–21
address-of 5–13
assignment 5–26, 17–7
bitwise 5–24
bitwise AND 5–24
bitwise exclusive OR 5–24
bitwise inclusive OR 5–25
cast 5–13, 19, 8–2
class member access 5–6
comma 5–27
conditional expression 5–25
conversion 12–5
copy assignment 12–19
decrement 5–7, 13, 15
division 5–21
equality 5–24
equality, comparison 5–23
example, scope resolution 10–5
function call 5–4, 13–19
function call 13–20
greater than 5–23
greater than or equal to 5–23
identities and overloading 13–20
increment 5–7, 13, 15
indirection 5–13
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 5–25
logical negation 5–13/14
modulus 5–21
multiplication 5–21
multiplicative 5–21
new —see new
one’s complement 5–13/14
overloaded 5–1
overloading —see also overloaded operator
overloading restrictions 13–20
pointer to member 5–20
precedence of 1–6
relational 5–23
right shift, right shift operator 5–22
scope resolution 5–2/3, 9–6, 10–1, 10
shift —see left shift operator, right shift operator
side effects and comma 5–27
side effects and logical AND 5–25
side effects and logical OR 5–25
sizeof 5–13, 15
subscripting 5–4, 13–19
unary 5–13/14
unary minus 5–13/14
unary plus 5–13/14
use, scope resolution 9–9
operator
bool::basic_ios 27–18
bool().basic_istream 27–33
bool().basic_ostream 27–42
delete 17–11, 18–11, 20–20
delete 5–19, 12–9
delete — see delete
delete[] 17–11, 18–12
delete[] 5–19, 12–9
function 13–19
new 17–11, 18–10/13, 20–20
new 5–17, 12–9
new[] 17–11, 18–12/13
new[] 5–17, 12–9
overloaded 13–19
operator:
  basic_ios 27–18
valarray 26–15
operator!= 20–5
  basic_string 21–25
  bitset 23–45
  complex 26–7
  istreambuf_iterator 24–24
locale 22–8
reverse_iterator 24–14
  type_info 18–14
valarray 26–18
operator=, valarray 26–17
operator+=
  gslice_array 26–24
  indirect_array 26–27
mask_array 26–26
slice_array 26–21
valarray 26–15
operator=
bitset 23–45
valarray 26–17
operator&&, valarray 26–17
operator&=
bitset 23–43
  gslice_array 26–24
  indirect_array 26–27
mask_array 26–26
slice_array 26–21
valarray 26–15
operator(), locale 22–8
operator*
  auto_ptr 20–19
  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 26–17
operator+=
complex 26–6
  gslice_array 26–24
  indirect_array 26–27
mask_array 26–26
slice_array 26–21
valarray 26–15
operator=
  basic_string 21–24
complex 26–6
reverse_iterator 24–13, 15
valarray 26–15, 17
operator++
back_insert_iterator 24–16
bad_alloc 18–13
bad_cast 18–15
bad_exception 18–17
bad_typeid 18–16
basic_string 21–14
exception 18–16
front_insert_iterator 24–17
gslice_array 26–24
indirect_array 26–27
insert_iterator 24–18
mask_array 26–25
ostreambuf_iterator 24–25
slice_array 26–21
type_info 18–15
valarray 26–14
operator==
    basic_string 21–25
    bitset 23–45
    complex 26–7
    istream_iterator 24–20
    istreambuf_iterator 24–24
    locale 22–8
    map 23–32
    multimap 23–35
    multiset 23–40
    pair 20–6
    queue 23–20
    reverse_iterator 24–14
    set 23–38
    type_info 18–14
    valarray 26–18
    vector 23–24
    vector<bool> 23–27
operator> 20–5
    basic_string 21–26
    reverse_iterator 24–14
    valarray 26–18
operator>>
    basic_string 21–27
    reverse_iterator 24–14
    valarray 26–18
operator>>
    basic_istream 27–34
    basic_string 21–27
    bitset 23–45
    complex 26–7
    istream 27–33
    valarray 26–17
operator==
    bitset 23–44
    gslice_array 26–24
    indirect_array 26–27
    mask_array 26–26
    slice_array 26–21
    valarray 26–15
operator[]
    basic_string 21–16
    map 23–32
    reverse_iterator 24–14
    valarray 26–14/15
operator~
    bitset 23–45
    valarray 26–17
operator^=
    bitset 23–43
    gslice_array 26–24
    indirect_array 26–27
    mask_array 26–26
    slice_array 26–21
    valarray 26–15
operator^=
    bitset 23–45
valarray 26–17
operator!=
    bitset 23–43
    gslice_array 26–24
    indirect_array 26–27
    mask_array 26–26
    slice_array 26–21
    valarray 26–15
operator|=
    bitset 23–44
    valarray 26–17/18
operator-
    bitset 23–44
    valarray 26–15
operator 13–19
operator-function-id 13–19
optimization of temporary — see elimination of temporary OR
    operator, bitwise exclusive 5–24
    operator, bitwise inclusive 5–25
    operator, logical 5–25
    operator, side effects and logical 5–25
order
    of argument evaluation 5–6
    of argument evaluation, unspecified 5–6
    of base class initialization 12–14
    of destruction of temporary 12–3
    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 12–7
    of execution, constructor and array 12–11
    of execution, constructor and static objects 12–12
    of execution, destructor 12–7
    of execution, destructor and array 12–7
    of execution, member constructor 12–2
    of execution, member destructor 12–7
    of function call evaluation, unspecified 5–6
    of initialization 3–22, 10–2
    of member initialization 12–14
    of virtual base class initialization 12–14
<ostream> 27–29
    ostream 27–4, 29
    ostreambuf_iterator 24–24
    failed 24–25
operator* 24–25
operator++ 24–25
operator+= 24–25
    ostreambuf_iterator 24–24
    ostreambuf_iterator 24–24
    ostream_iterator 24–20
    ostringstream 27–4, 48
    ostrstream D–10
    :pcount D–10
    freeeze D–10
    ostringstream D–10
    ostrstream D–10
    rdbuf D–10
    str D–10
    out, codevar 22–17
    out_of_range 19–3, 21–9, 23–42/45
    out_of_range 19–3
    output_iterator_tag 24–9/10
    overflow 5–1
    undefined 5–1
    overflow
    basic_filebuf 27–59
    basic_stringbuf 27–28
    basic_stringbuf 27–30
overloading 8
overloaded
overload
overload keyword anachronism C
overrider, final 10
pair 20–6
operator== 20–6
operator-- 20–6
parameter 1–3, 8–15
adjustment, array 8–10
adjustment, function 8–10
declaration 8–9
element, 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 3–21
parentheses
and ambiguity 5–16
in declaration 8–3, 5
parenthesized expression 5–3
partial specializations, class template 14–12
partial_sort 25–19
partial_sort_copy 25–19
partial_sum 26–29
partition 25–17
pbackfail
basic_filebuf 27–59
basic_streambuf 27–28
basic_stringbuf 27–50
strstreambuf D–7
pbase, basic_streambuf 27–25
pbump, basic_streambuf 27–25
:pcount, ostrstream D–10
pcount
strstream D–12
strstreambuf D–6
peek, basic_istream 27–37
period 17–6
phases, translation 2–1
placement
of object, destructor and 12–8
syntax, new 5–17
plus 20–9
pm-expression 5–20
POD
class type 5–17
type 3–31
type 5–17
POD-struct 9–1
point
of declaration class name 9–3
of declaration, enumerator 3–5
of declaration name 3–5
of definition, enumerator 7–11
of error checking 14–17
of instantiation 14–25
promotion, floating 4–4
type, floating 3–31
pointer
— see also void *
and pointer to member type, multi-level mixed 4–3
arithmetic 5–22
cast, integer to 5–11
comparison 5–23
comparison, undefined 5–22/23
comparison, unspecified 5–23
comparison, void* 5–23
constant, null 4–4/5

P
pair 20–6
operator< 20–6

DRAFT: 2 December 1996 Index 25
S
sbumpc, basic_streambuf 27–24
scalar type 3–31
scan_is
cctype 22–11
cctype<uchar> 22–15
scan_not
cctype 22–11
cctype<uchar> 22–15
s-char 2–11
s-char-sequence 2–10
scientific 27–20
scope 3–1, 4
anonymous union at namespace 9–10
class 3–7
destructor and exit from 6–5
extension declaration 3–6
file 17–10
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 12–10
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 class anachronism C–13
of nested type name 9–13
of parameter 3–6
overloading and 13–3
potential 3–4
resolution operator 5–2, 6–5, 10–1, 10
resolution operator :: 3–13
resolution operator example 10–5
resolution operator use 9–9
rules summary 3–8
selection-statement 6–2
scoping
ambiguity resolution 10–5
and new 5–16
search 25–12
seekdir, ios_base 27–11
seekg, basic_istream 27–38
seekoff
basic_filebuf 27–60
basic_streambuf 27–26
basic_stringbuf 27–50
strstreambuf D–7
seekp, basic_ostream 27–42
seekpos
basic_filebuf 27–60
basic_streambuf 27–26
basic_stringbuf 27–51
strstreambuf D–8
selection statement 6–2
selection-statement 6–2
scope 6–2
semantics, class member 5–6
sentry
See also constructor, destructor, template explicit 14

see also constructor, destructor, template 14
class template 14

template specializations, class template partial 14–12
streampos, implementation-defined D–2
streamsize 27–7
strtime 22–35
stride
 الغربية 26–23
_slice 26–20
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 17–6
null-terminated wide-character 17–6
sizeof 2–11
terminator 0 2–11
type of 2–11
<string> 21–6
<string.h> D–1
string-literal 2–10
stringstream 27–4
strlen D–5, 10
strpbrk 21–30
strrchr 21–30
strrstr 21–30
stringstream D–11
pcnt D–12
rdbuf D–11
str D–12
stringstream D–11
stringstream D–11
~stringstream D–11
~stringstream, stringstream D–11
stringstreambuf D–3
freeze D–6
overflow D–6
pbackfail D–7
pcnt D–6
seekoff D–7
seekpos D–8
setbuf D–9
str D–6
stringstreambuf D–5
stringstreambuf D–5
underflow D–7
~stringstreambuf D–6
~stringstreambuf, stringstreambuf 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, 13–19
operator, overloaded 13–21
subsequence rule, overloading 13–16
_substr, basic_string 21–23
subtraction
implementation defined pointer 5–22
operator 5–21
suffix
E 2–10
F 2–10
L 2–8, 10
U 2–8
f 2–10
1 2–8, 10
u 2–8
sum, valarray 26–16
summary
compatibility with C C–1
compatibility with ISO C C–2
scope rules 3–8
syntax A–1
swap 25–13
basic_string 21–20, 27
swap_ranges 25–13
switch
statement 6–2/3, 6
statement, declaration in 6–3
.sync
basic_filebuf 27–61
basic_istream 27–38
basic_stringbuf 27–26
sync_with_stdio, ios_base 27–13
synonym 7–15
type name as 7–5
tsyntax
checking 14–17
class member 5–6
notation 1–3
summary A–1

table, _ctype<char> 22–15
tan 26–19, 31
complex 26–9
tanh 26–19, 31
complex 26–9
tellg, basic_istream 27–38
tellp, basic_ostream 27–42
template 14–1
and < 14–4
and friend 14–11
and static member 14–9
argument 14–5
argument deduction 14–36
argument specification 14–35
class 23–42
definition of 14–1
explicit specialization 14–31
function 14–34
implicit instantiation 14–27
member function 14–8
name lookup 14–16
name resolution 14–16
overload resolution 14–16
overload resolution, function 14–41
partial specializations, class 14–12
primary 14–12
specialization 14–27
specialization, class 14–5
type equivalence 14–7
template 14–1
template-argument 14–4
template-argument-list 14–4
template-declaration 14–1
Index

template-id 14–3

template-name 14–3

template-parameter 14–2

template-parameter-list 14–1

temporary 12–2

and declarator initializer 12–3

destructor for 12–3

elimination of 12–2

implementation-defined generation of 12–2

order of destruction of 12–3

terminate 3–23, 15–8, 18–9, 18

terminate() 15–8

terminate_handler 17–11, 18–18

termination

and destructor, program 12–8

program 3–22/23

terminator 0, string 2–11

terminology, pointer 3–33

test, bitset 23–45

this 5–2

anachronism, assignment to C–12

and constructor anachronism C–12

and destructor anachronism C–12

pointer — see this

type of 9–7

thousands_sep, numpunct 22–29

throw 15–1

throw-expression in conditional-expression 5–25

throw-expression 15–1

throwing, exception 15–2

tie, basic_ios 27–17

time_get 22–32

date_order 22–33

do_date_order 22–33

do_get_date 22–34

do_get_monthname 22–34

do_get_time 22–33

do_get_weekday 22–34

do_get_year 22–34

generate 22–33

generate_get_monthname 22–33

generate_get_time 22–33

generate_get_weekday 22–33

generate_get_year 22–33

time_get_byname 22–34

<time.h> 22–34

time_get_byname 22–34

<time.h> D–1

time_put 22–35

do_put 22–35

put 22–35

time_put_byname 22–35

times 20–9

to

int, bool promotion 4–3

rvalue, lvalue conversion 4–2

token 2–4, 7

token 2–4

tolower 22–10

type 22–11

type_char 22–15

to_string, bitset 23–44

to_ulong, bitset 23–44

toupper 22–10

type 22–11

type_char 22–15

transform 25–13

collate 22–31

translation

phases 2–1

separate 2–1

unit 17–9/10

unit 2–1, 3–19

unit, name and 3–1

tigraph 2–1/2

truename, false_name, numpunct 22–29

truncation 4–4

try 15–1

try-block 15–1

type 3–1

Boolean 3–31

POD 3–31

ambiguity, declaration 7–2

arithmetic 3–32

array 3–32, 8–10

bitmask 17–5

char 3–31

class 3–31

checking, argument 5–5

checking, extension to C C–1

checking of default argument 8–12

class and 9–1

completely defined object 9–4

compound 3–32

const 7–6

conversion, explicit — see casting

declaration 8–4

declaration consistency 3–21

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, type 14–7

example of incomplete 3–30

extension to C reference C–1

extension to C user-defined C–1

float 3–32

floating point 3–31

function 3–32, 8–9/10

fundamental 3–31

generator — see template

implementation-defined sizeof 3–31

incomplete 3–2/3, 5, 30, 4–2, 5–4/6, 8/9, 14/15, 19, 21, 26, 10–1

int 3–31

integral 3–31

long 3–31

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 12–5

of delete 12–10

of enum 7–10/11

of floating point literal 2–10

of integer literal 2–8

of integer literal, implementation-defined 2–8

of new 12–9

of ptrdiff_t, implementation defined 5–22
class versus 9–1
constructor 9–10
default member access 9–1
destructor 9–10
extension to C anonymous C–1
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 25–15
list 23–19
unique_copy 25–16
unit
instantiation 2–2
translation 17–9/10
unitbuf 27–20
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 1–9
unqualified-id 5–3
unrepresentable integral value, undefined 4–4
unsetf, ios_base 27–12
unshift, codecvt 22–17
unsigned
arithmetic 3–32
char type 3–31/32
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 18–11/12, 14, 21–12, 25–19, 26–16, 27–50/D–7
address of member function 17–13
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 5–23
unwinding, stack 15–4
up, name look 3–8
upper_bound 25–20
uppercase 17–6, 10
uppercase 27–19
use_facet, locale 22–9
user-defined
conversion 12–4/5
conversion and name hiding 12–6
conversion, implicit 12–6
conversion, inheritance of 12–6
type, extension to C C–1
using-declaration 7–16
using-directive 7–21
usual arithmetic conversions 5–2
utility> 20–5
va_end 17–10
<valarray> 26–9
valarray 26–12, 24
apply 26–17
cshif 26–17
length 26–16
max 26–16
min 26–16
operator! 26–15
operator!= 26–18
operator% 26–17
operator%= 26–15
operator&& 26–17
operator&&= 26–15
operator+ 26–15, 17
operator= 26–15
operator/= 26–17
operator/= 26–15
operator<< 26–18
operator<<= 26–15
operator<<= 26–18
operator<<= 26–15
operator<<= 26–18
operator<<= 26–15
operator<<= 26–18
operator<<= 26–15
operator<<= 26–18
operator<<= 26–15
operator<<= 26–18
operator<< 26–16
sum 26–16
<valarray> 26–13
valarray 26–13
<~valarray> 26–14
~valarray, valarray 26–14
va_list 17–10
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> 23–10
vector 23–22
assign 23–24
capacity 23–25
erase 23–25
insert 23–25
operator< 23–24
operator== 23–24
reserve 23–25
resize 23–25
vector 23–24
vector< bool> 23–26
operator< 23–27
operator== 23–27
virtual 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 12–14, 20
base class initialization, order of 12–14
destructor 12–7
destructor, pure 12–7
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, 13
user-defined conversion 12–6
virtual
and friend 10–8
and multiple inheritance 10–8
specifier 7–5
visibility 3–8
void
parameter 8–9
type 3–32
type specifier 7–8
voids 8–6
void*
conversion, pointer to member 4–5
pointer comparison 5–23
pointer type extension to C C–1
type 3–33
volatile 3–33
constructor and 9–8, 12–1
destructor and 9–8, 12–7
extension to C C–2
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
wchar_t type-specifier 7–8
wchar_t 2–9, 11, 17–6, 21–30
implementation-defined 3–32
type 3–32
underlying type 3–32
wcin 27–6
wclog 27–6
wcout 27–6
wcschr 21–30
wcsrchr 21–31
wcsstr 21–31
<wctype.h> D–1
well-formed program 1–3
wfilebuf 27–4, 56
wifstream 27–4
what
bad_alloc 18–13
bad_cast 18–15
bad_exception 18–17
bad_typeid 18–16
exception 18–17
ios_base::failure 27–9
white
space 2–3
space 2–4
wide string literal 2–11
wide-character 2–9
string, null-terminated 17–6
widen
basic_ios 27–17
cctype 22–11
cctype<char> 22–15
width, ios_base 22–10, 27–13
wifstream 27–4, 56
wios 27–7
wiostream 27–4, 29
wistringstream 27–4, 48
wmemchr 21–31
wofstream 27–4, 56
wostream 27–4, 29
wstringstream 27–4, 48
write, basic_ostream 27–45
ws 27–34, 39
wstreambuf 27–4, 21
wstreamoff 27–7
wstringbuf 27–4, 48
wstringstream 27–4
X
xalloc, ios_base 27–13
xsgetn, basic_streambuf 27–26
xsgetn, basic_streambuf 27–28
X(X&) — see copy constructor .19
Z
zero
pointer 4–4
undefined division by 5–1, 21
undefined modulus 5–1
width of bit-field 9–11
zero-initialization 8–16