Index

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

A

abort() 3—4, 15—5
abstract
 class 10—8
 class, constructor and 10—9
 class, pointer to 10—9
abstract-declarator 8—2
access
 adjusting base class member 11—3
 ambiguity, member 10—4
 and friend, class 11—6
 and friend function 11—5

base class 11—2
base class member 10—1
class member 5—4
control 11—1
control, anonymous `union` 9—9
control default 11—1
control, member function and 12—1
control, overloading resolution and 10—4
declaration 11—3
declaration, overloaded name and 11—5
default assignment operator 12—12
default copy constructor 12—12
example, member name 11—3
member name 11—1
overloading and 13—3
`protected` member 11—7
specifier 11—1, 11—2
specifier and `friend` 11—6
specifier and object layout 11—2
`struct` default member 9—1
`union` default member 9—1
virtual function 11—7
access-specifier 10—1
addition operator 5—11
additive operator 5—11
additive-expression 5—11
address
of bit-field 9—10
of bit-field restriction 9—10
of constructor 12—2
of overloaded function 5—6, 13—6
of qualified name 5—6
address-of operator 5—6
adjusting base class member access 11—3
adjustment
array parameter 8—7
function parameter 8—7
aggregate 8—11
alert 2—7
alignment
of array 5—10
of bit-field 9—10
of bit-field, implementation dependency 9—10
of class 5—10
of class member 5—10
requirement, implementation dependency 3—5
restriction 5—10
allocation
function 12—6
implementation dependency 9—4, 11—2
implementation dependency base class 10—2
implementation dependency bit-field 9—10
`new`, storage 5—7
ambiguity
base class member 10—4
class conversion 10—5
declaration type 7—1
declaration versus cast 8—3
declaration versus expression 6—6
detection, overloaded function 13—3
function declaration 8—11
`if-else` 6—3
member access 10—4
parentheses and 5—8
pointer conversion 4—3
pointer to member conversion 4—3
reference conversion 4—3
resolution, scoping 10—4
anachronism 19—12
C function definition 19—12
assignment to `this` 19—13
cast of pointer to member 19—13
free store and constructor 19—13
free store and destructor 19—13
memory management 19—13
nonnested class 19—14
old style base class initializer 19—13
old style function definition 19—12
`overload` keyword 19—12
pointer to member conversion 19—13
scope of nested class 19—14
`this` and constructor 19—13
`this` and destructor 19—13
AND
operator, bitwise 5—13
operator, logical 5—14
operator, side effects and logical 5—14
anonymous
`union` 9—9
`union` access control 9—9
`union`, extension to C 19—2
`union`, global 9—9
`union` restriction 9—9
ANSI
C headers 2—5
C summary, compatibility with 19—2
`argc` 3—3
argument 1—3
class object as 12—9
conversion 5—4, 8—7
evaluation, order of 5—4
evaluation, unspecified order of 5—4
list, empty 8—7
list, variable 8—7
matching − see overloading resolution
passing 5—3
passing, reference and 8—13
reference 5—3
substitution 16—5
to constructor, unspecified 5—8
type checking 5—3
type conversion 12—3
type, unknown 8—7
`argv[]` 3—3
arithmetic
conversion 4—2
exception 5—1
exception, implementation dependency 5—1
extension to C single precision 19—2
pointer 5—11, 5—12
single precision floating point 4—2
type 3—5
`unsigned` 3—5
array
alignment of 5—10
`const` 7—5
constructor and 5—8
declaration 8—6
declarator `[]` 8—6
declarator, multidimensional 8—6
default constructor and 5—8
example 8—6
initialization 8—11
member 9—4
multidimensional 8—6
`new` 5—8
of class objects and constructor 12—9
of class objects and default constructor 12—9
of class objects and `new` 5—8
of class objects initialization 8—12, 12—9
order of execution, constructor and 12—2
order of execution, destructor and 12—5

overloading and pointer versus 13—2
parameter adjustment 8—7
pointer conversion 4—3
size, default 8—6
`sizeof` 5—7
storage of 8—7
type 3—6
arrow operator − see class member access operator
`asm`
declaration 7—8
implementation dependency 7—8
assembler 7—8
assignment
and initialization, overloaded 12—9
and lvalue 5—15
base class object 5—15
`const` pointer 5—15
conversion by 5—15
derived class object 5—15
expression 5—15
extension to C memberwise 19—2
member 12—12
memberwise 13—7
of class object 12—13
of derived class to base class 12—13
operator 5—15, 12—12
operator access, default 12—12
operator, default 13—7
operator, default 12—12, 12—13
operator, overloaded 13—7
operator restriction, default 12—12
pointer to `const` 5—15
pointer to member 5—15
pointer to `volatile` 5—15
reference 8—13
to class object 5—15
to pointer 5—15
to pointer to member 5—15
to pointer to member, zero 5—15
to pointer, zero 5—15
to reference 5—15
to `this` anachronism 19—13
`volatile` pointer 5—15
assignment-expression 5—15
assignment-operator 5—15
`atexit()` 3—4
`auto`
destruction of 6—5, 6—6
initialization 6—6
object initialization 8—10
restriction 7—2
specifier 7—2
storage class 3—4

B

backslash character 2—7
backspace 2—7
base
class 10—1, 10—3
class access 11—2
class allocation, implementation dependency 10—2
class, assignment of derived class to 12—13
class cast 5—10
class constructor order of execution 12—2
class destructor order of execution 12—5
class, direct 10—1
class, indirect 10—1
class initialization 12—9, 12—10
class initialization, order of 12—10
class initializer 8—10
class initializer anachronism, old style 19—13
class member access 10—1
class member access, adjusting 11—3
class member ambiguity 10—4
class object, assignment 5—15
class pointer conversion 4—3
class, `private` 11—2
class, `public` 11—2
class, reference to 4—3
class virtual − see virtual base class
of integer literal 2—6
base-specifier 10—1
base-specifier-list 10—1
Ben 13—2
binary
operator, interpretation of 13—7
operator, overloaded 13—7
binding
− see virtual function, dynamic
default parameter 8—9
bit-field 9—10
address of 9—10
alignment of 9—10
allocation, implementation dependency 9—10
declaration 9—10
implementation dependency alignment of 9—10
implementation dependency sign of 9—10
layout 9—10
restriction 9—10
restriction, address of 9—10
restriction, pointer to 9—10
type of 9—10
unnamed 9—10
zero width of 9—10
bitwise
AND operator 5—13
exclusive OR operator 5—14
inclusive OR operator 5—14
operator 5—13
block
initialization in 6—6
scope − see local scope
statement `{}` 6—2
structure 6—5
body, function 8—10
bound pointer to member function, undefined 19—13
`break` statement 6—5
built-in type − see fundamental type
byte 5—7

C

C
anonymous `union`, extension to 19—2
class, extension to 19—2
`const`, extension to 19—2
dangerous extension to 19—12
declaration statement, extension to 19—2
`delete`, extension to 19—2
destructor, extension to 19—2
expression evaluation, difference from 19—1
extension to 1—1, 19—1, 19—2
function definition anachronism 19—12
headers, ANSI 2—5
implementation dependency extension to 19—12
inline function, extension to 19—2
linkage to 7—8
memberwise assignment, extension to 19—2
memberwise initialization, extension to 19—2

multiple inheritance, extension to 19—2
`new`, extension to 19—2
overloading `delete`, extension to 19—2
overloading, extension to 19—2
overloading `new`, extension to 19—2
pointer to member, extension to 19—2
`protected`, extension to 19—2
reference type, extension to 19—2
scope, difference from 19—1
single precision arithmetic, extension to 19—2
`sizeof`, difference from 19—1
summary, compatibility with 19—1
summary, compatibility with ANSI 19—2
type checking, extension to 19—1
user-defined type, extension to 19—2
`void*` pointer type extension to 19—2
`volatile`, extension to 19—2
C++, evolution of 1—1
call
− see also function call, member function call, overloaded function call, virtual function call
by reference 5—3
by value 5—3
operator function 13—7
carriage return 2—7
`case` label 6—1, 6—3
cast
ambiguity, declaration versus 8—3
base class 5—10
class object 5—10
`const` 5—10
derived class 5—10
dynamic 5—5
implementation dependency pointer to function 5—10
integer to pointer 5—9
lvalue 5—10
of pointer to member anachronism 19—13
operator 5—6, 5—9, 8—2
pointer to function 5—10
pointer to integer 5—9
pointer to member 5—10
pointer to pointer 5—10
reference 5—10
to undefined class 5—10
cast-expression 5—9
casting 5—4, 5—9
pointer to `volatile` 5—10
`catch` 15—1
c-char 2—6
c-char-sequence 2—6
change
to `const` object, implementation dependency 5—10
to `const` object, undefined 7—5
to string literal, undefined 2—8
`char`
implementation dependency sign of 3—5
integer conversion 4—1
literal, implementation dependency value of 2—7
type 3—5
type, `signed` 3—5
type specifier 7—6
type, `unsigned` 3—5
character
array initialization 8—13
constant 2—7
literal 2—7
literal, type of 2—7
`signed` 3—5
string 2—8
character-literal 2—6
class 3—6, 9—1
abstract 10—8
access and `friend` 11—6
alignment of 5—10
anachronism, nonnested 19—14
and type 9—1
base − see base class
cast to undefined 5—10
constructor and abstract 10—9
conversion 12—3
conversion ambiguity 10—5
declaration, forward 9—2, 10—1
declaration syntax summary 18—9
declaration `{}` 9—1
definition 9—1, 9—3
definition 3—2
definition example 9—4
definition name hiding 9—2
definition, scope of 9—2
definition `{}` 9—1
derived − see derived class
extension to C 19—2
`friend` 11—6
lattice − see DAG
linkage of 3—3
linkage specification 7—8
local − see local class
member − see also member
member access 5—4
member access operator 5—4
member, alignment of 5—10
member declaration 9—3
member function 9—6
member initialization 8—11
member semantics 5—4
member, `static` 3—4
member storage class 3—4
member syntax 5—4
name 8—2
name as type definition 9—2
name declaration 3—1
name, elaborated 7—6, 9—2, 9—3
name, point of declaration 9—3
name, scope of 9—2
name, `typedef` 7—4, 7—5, 9—3
nested − see nested class
object as argument 12—9
object, assignment of 12—13
object, assignment to 5—15
object cast 5—10
object, `const` 7—5, 9—7
object copy 12—12
object copy − see also copy constructor
object copy example 12—13
object initialization 8—11, 12—8
object initialization − see also constructor
object layout 9—4, 10—3
object, member 9—4
object, operations on 9—1
object return type 12—9
object, `sizeof` 5—7
objects and constructor, array of 12—9
objects and default constructor, array of 12—9
objects and `new`, array of 5—8
objects initialization, array of 8—12, 12—9
pointer to abstract 10—9
polymorphic 10—6
scope 3—2
scope of enumerator 7—8
`sizeof`, empty 9—1
template 14—2
type restriction, member of 12—9

unnamed 7—5
class
type specifier 7—6
versus struct 9—1
versus union 9—1
class-key 7—6, 9—1
class-name 9—1
class-specifier 9—1
comma
operator 5—16
operator, side effects and 5—16
comment 2—3
/* */ 2—4
// 2—4
comparison
implementation dependency pointer 5—13
pointer 5—13
pointer to function 5—13
undefined pointer 5—12, 5—13
void* pointer 5—13
compatibility
with ANSI C summary 19—2
with C summary 19—1
compilation, separate 2—1
compiler control line − see preprocessing directive
complete object 1—2
compound statement 6—2
compound-statement 6—2
concatenation
string 2—8
undefined string literal 2—8
condition 6—2
conditional
expression operator 5—14
inclusion 16—2
conditional-expression, throw-expression in 5—14
*condition*s, rules for 6—2
consistency
example, linkage 7—2
linkage 3—3, 7—2
linkage specification 7—9
type declaration 3—3
*const example 8—4
const 3—6
array 7—5
assignment, pointer to 5—15
cast 5—10
class object 7—5, 9—7
constructor and 9—8, 12—1
destructor and 9—8, 12—5
example 8—4
extension to C 19—2
initialization 7—5, 8—11
initialization, pointer to 8—11
linkage of 3—3, 7—2
member function 9—7
member initialization 12—10
object, implementation dependency change to 5—10
object, undefined change to 7—5
operand 5—1
overloading and 13—1
pointer assignment 5—15
pointer initialization 8—11
reference 8—13
type 7—5
void* pointer conversion 4—3
volatile void* pointer conversion 4—3
constant 2—5, 3—6, 5—2
character 2—7
enumeration 7—7
expression 5—16
expression, pointer to member 5—6
integer 2—6
long 2—6
pointer declaration 8—4
pointer example 8—4
unsigned 2—6
constant-expression 5—16
construction, order-of 3—4
constructor 12—1
access, default copy 12—12
address of 12—2
anachronism, free store and 19—13
anachronism, this and 19—13
and abstract class 10—9
and array 5—8
and array order of execution 12—2
and const 9—8, 12—1
and initialization 12—8
and initialization example 12—8
and member function 12—2
and member function call 12—11
and new 5—8
and new, implementation dependency 5—8
and return 6—5
and static objects order of execution 12—9
and virtual function call 12—11
and volatile 9—8, 12—1
array of class objects and 12—9
call, explicit 12—2
conversion by 12—3
conversion by − see also user-defined conversion
copy 12—1, 12—2, 12—12
default − see default constructor
default copy 12—12, 12—13
definition 8—10
example 12—2
exception handling 15—3
for temporary 12—2
inheritance of 12—1
local object 3—4
order of execution, base class 12—2
order of execution, member 12—2
restriction 12—1, 12—2
restriction, default copy 12—12
type of 12—2
union 9—9
unspecified argument to 5—8
continue
in for statement 6—4
statement 6—5
control line − see preprocessing directive
conversion
− see also type conversion
ambiguity, class 10—5
ambiguity, pointer 4—3
ambiguity, pointer to member 4—3
ambiguity, reference 4—3
anachronism, pointer to member 19—13
and name hiding, user-defined 12—5
argument 5—4, 8—7
arithmetic 4—2
array pointer 4—3
base class pointer 4—3
by assignment 5—15
by constructor 12—3
char integer 4—1
class 12—3
const void* pointer 4—3
const volatile void* pointer 4—3
derived class pointer 4—3
explicit type − see casting

floating point integer 4—2
function − see also user-defined conversion
implementation dependency floating point 4—2
implementation dependency integer 4—1
implementation dependency pointer integer 5—9
implicit 4—1, 5—2, 12—3
implicit user-defined 12—4
inheritance of user-defined 12—4
integer 4—1
lvalue 4—1
null pointer 4—3
operator 5—2, 12—4
out of range value, undefined 4—2
overloaded function and standard 13—4
overloading resolution and 13—3
overloading resolution and pointer 13—6
overloading resolution and standard 13—4
overloading resolution and user-defined 13—4, 13—5
pointer 4—2
pointer to function 4—3
pointer to member 4—3
pointer to member `void*` 4—4
reference 4—3
return type 6—5
rules, type 4—2
safe floating point 4—2
`signed` `unsigned` integer 4—1
standard 4—1
type of 12—4
user-defined 5—2, 12—3, 12—4
virtual user-defined 12—4
`void*` pointer 4—3
`volatile` `const` `void*` pointer 4—3
`volatile` `void*` pointer 4—3
zero pointer 4—3
conversion-function-id 12—4
conversions
cv-qualifier pointer 4—2
cv-qualifier reference 4—3
copy
class object 12—12
constructor 12—1, 12—2, 12—12
constructor access, default 12—12
constructor, default 12—12, 12—13
constructor restriction, default 12—12
example, class object 12—13
ctor-initializer 12—9
cv-qualifier 3—6
pointer conversions 4—2
reference conversions 4—3
cv-qualifier 8—2

D

DAG
multiple inheritance 10—3
nonvirtual base class 10—3
virtual base class 10—3
dangerous extension to C 19—12
data member − see member
deallocation − see `delete`
decimal literal 2—6
decimal-literal 2—5
decimal-point definition 17—9
declaration 3—1, 7—1
`:`, field 9—10
access 11—3
ambiguity, function 8—11
array 8—6
as definition 7—2
`asm` 7—8
bit-field 9—10
class member 9—3
class name 3—1
class name, point of 9—3
consistency, type 3—3
constant pointer 8—4
default parameter 8—8
definition versus 3—1
ellipsis in function 5—4, 8—7
enumerator, point of 3—2
example 3—1, 8—8
example, function 8—8
`extern` 3—1
`extern` reference 8—13
forward 7—3
forward class 9—2, 10—1
function 3—1, 8—7
function member 9—6
function template 14—4
hiding − see name hiding
in `for`, scope of 6—4
in `for` statement 6—4
in `switch` statement 6—3
matching, overloaded function 13—2
member 9—3
multiple 3—3
name 3—1
name, point of 3—2
overloaded name and access 11—5
overloaded name and `friend` 11—5
parameter 8—7
parentheses in 8—3
pointer 8—4
reference 8—4
`register` 7—2
specifier 7—1
statement 6—5
statement, extension to C 19—2
`static` member 3—1
storage class 7—2
syntax summary 18—6
syntax summary, class 18—9
template 14—4
template class 14—4
template function 14—4
type 8—3
type ambiguity 7—1
`typedef` 3—1
`typedef` as type 7—4
versus cast ambiguity 8—3
versus expression ambiguity 6—6
`{}`, class 9—1
`{}`, `enum` 7—6
declaration 7—1
declaration-seq 7—8
declaration-statement 6—5
declarator 7—1, 8—1
`&`, reference 8—4
`()`, function 8—7
`*`, pointer 8—4
`::*`, pointer to member 8—5
`[]`, array 8—6
example 8—2
initializer, temporary and 12—3
meaning of 8—3
multidimensional array 8—6
syntax summary 18—8
declarator 8—1
declarator-id 8—2
decl-specifier 7—1

decrement
operator 5—5, 5—6, 5—7
operator, overloaded 13—8
default
access control 11—1
argument, overloading resolution and 13—3
array size 8—6
assignment operator 13—7
assignment operator 12—12, 12—13
assignment operator access 12—12
assignment operator restriction 12—12
constructor 12—1, 12—9, 12—12
constructor and array 5—8
constructor and initialization 12—8
constructor and `new` 5—8
constructor, array of class objects and 12—9
copy constructor 12—12, 12—13
copy constructor access 12—12
copy constructor restriction 12—12
destructor 12—5
initialization 8—11
member access, `struct` 9—1
member access, `union` 9—1
parameter and name hiding 8—9
parameter binding 8—9
parameter declaration 8—8
parameter evaluation 8—9
parameter example 8—8, 8—9
parameter, overloaded operator and 13—7
parameter, overloading and 8—10
parameter, scope of 8—9
parameter, temporary and 12—3
parameter type checking 8—9
type `int` 7—5
`default` label 6—1, 6—3
`#define` 16—5
definition
NTCS 17—8
and initialization 7—2
class 3—2
class 9—1, 9—3
class name as type 9—2
constructor 8—10
decimal-point 17—9
declaration as 7—2
enumerator 3—2
enumerator point of 7—7
example 3—1
example, function 8—10
example, nested class 9—10, 9—11
function 3—2
function 8—10
function template 14—4
inline member function 3—3
letter 17—9
local class 9—11
member 9—7
member function 9—6, 9—7, 9—8
name hiding, class 9—2
nested class 9—10
object 3—2
pure virtual function 10—8
scope, macro 16—6
scope of class 9—2
scope of function 3—3
`static` member 9—9
string 17—8
template class 14—4
template function 14—4
versus declaration 3—1
virtual function 10—7
`{}`, class 9—1
`delete` 5—9, 12—7
destructor and 5—9, 12—5
example 12—7
example, destructor and 12—8
example, scope of 12—8
extension to C 19—2
extension to C overloading 19—2
overloading and 12—7
type of 12—7
undefined 5—9
undefined value 5—9
deleted object, undefined 5—9
delete-expression 5—9
dereferencing 5—2
− see also indirection
derivation − see inheritance
derived
class 10—1
class cast 5—10
class example 10—2
class, most 12—10
class object, assignment 5—15
class, overloading and 13—2
class pointer conversion 4—3
class to base class, assignment of 12—13
type 3—6
destruction
of `auto` 6—5, 6—6
of local `static` 6—6
of local variable 6—5, 6—6
of temporary 12—3
of temporary, order of 12—3
order-of 3—4
destructor 12—5
anachronism, free store and 19—13
anachronism, `this` and 19—13
and array order of execution 12—5
and `const` 9—8, 12—5
and `delete` 5—9, 12—5
and `delete` example 12—8
and exit from scope 6—5
and fundamental type 12—6
and member function 12—5
and member function call 12—11
and placement of object 12—6
and `static` objects order of execution 12—9
and virtual function call 12—11
and `volatile` 9—8, 12—5
call example, explicit 12—6
call, explicit 12—5, 12—6
call, implicit 12—5
call, unspecified 6—6
default 12—5
exception handling 15—3
extension to C 19—2
for temporary 12—2
inheritance of 12—5
local object 3—4
order of execution 12—5
order of execution, base class 12—5
order of execution, member 12—5
program termination and 12—5
pure virtual 12—5
restriction 12—5
`static` object 3—4
`union` 9—9
virtual 12—5
difference
from C expression evaluation 19—1
from C scope 19—1

from C `sizeof` 19—1
digit 2—4
digit-sequence 2—8
digraph 2—3, 2—5
direct base class 10—1
direct-abstract-declarator 8—2
direct-declarator 8—2
directed acyclic graph − see DAG
directive
error 16—8
null 16—8
pragma 16—8
preprocessing 16—1
direct-new-declarator 5—7
distinct string 2—8
division
by zero, undefined 5—1, 5—11
implementation dependency 5—11
operator 5—11
`do` statement 6—3, 6—4
dominance, virtual base class 10—5
dot operator − see class member access operator
double quote 2—7
`double`
literal 2—8
type 3—5
type specifier 7—6
dynamic
binding − see virtual function
cast 5—5
initialization 3—4
type 1—3

E

`E` suffix 2—8
elaborated
class name 7—6, 9—2, 9—3
`enum` name 7—6
type specifier − see elaborated class name
elaborated-type-specifier 7—6
`#elif` 16—3
elimination of temporary 12—2
ellipsis
example 8—8
in function declaration 5—4, 8—7
overloading resolution and 13—3, 13—4
`#else` 16—3
`else` 6—2
empty
argument list 8—7
class `sizeof` 9—1
statement 6—2
`#endif` 16—3
`enum`
declaration `{}` 7—6
name, elaborated 7—6
overloading and 13—2
type of 7—6, 7—7
type specifier 7—6
enumerated type 3—5
enumeration 7—6
constant 7—7
example 7—7
underlying type 7—7
enumerator
class, scope of 7—8
definition 3—2
linkage of 3—3
member 7—8
point of declaration 3—2
point of definition 7—7
redefinition 7—7
restriction 7—7
value of 7—7
enumerator 7—7
environment, program 3—3
equality operator 5—13
equality-expression 5—13
equivalence
template type 14—3
type 7—4, 9—2
error directive 16—8
`#error` 16—8
escape
character − see backslash
sequence 2—7
sequence, undefined 2—7
escape-sequence 2—6
evaluation
default parameter 8—9
difference from C expression 19—1
`new`, unspecified order of 5—8
of expression, order of 5—1
order of argument 5—4
unspecified order of 5—1
unspecified order of argument 5—4
unspecified order of function call 5—4
evolution of C++ 1—1
exact match, overloading resolution 13—4
example
`*const` 8—4
array 8—6
class definition 9—4
class object copy 12—13
`const` 8—4
constant pointer 8—4
constructor 12—2
constructor and initialization 12—8
declaration 3—1, 8—8
declarator 8—2
default parameter 8—8, 8—9
definition 3—1
`delete` 12—7
derived class 10—2
destructor and `delete` 12—8
ellipsis 8—8
enumeration 7—7
explicit destructor call 12—6
explicit qualification 10—4
`friend` 9—2
`friend` function 11—5
function declaration 8—8
function definition 8—10
incomplete type 3—5
linkage consistency 7—2
local class 9—11
member function 9—6, 11—5
member name access 11—3
nested class 9—10
nested class definition 9—10, 9—11
nested class forward declaration 9—11
nested type name 9—12
overloading 13—1
pointer to member 8—5
pure virtual function 10—8
scope of `delete` 12—8
scope resolution operator 10—4
`static` member 9—8
subscripting 8—6
type name 8—2

typedef 7—4
unnamed parameter 8—10
variable parameter list 8—8
virtual function 10—6, 10—7
exception
arithmetic 5—1
handler 15—3
handling 15—1
handling constructor 15—3
handling destructor 15—3
handling syntax summary 18—12
implementation dependency arithmetic 5—1
throwing 15—2
exception-declaration 15—1
exception-specification 15—4
exit from scope, destructor and 6—5
exit() 3—3, 3—4
explanation, subscripting 8—6
explicit
constructor call 12—2
destructor call 12—5, 12—6
destructor call example 12—6
qualification example 10—4
type conversion − see casting
exponent-part 2—8
expression 5—1
ambiguity, declaration versus 6—6
assignment 5—15
constant 5—16
evaluation, difference from C 19—1
order of evaluation of 5—1
parenthesized 5—2
pointer to member constant 5—6
postfix 5—3
primary 5—2
reference 5—2
statement 6—2
syntax summary 18—1
unary 5—6
unspecified 5—4
expression 5—16
expression-list 5—3
expression-statement 6—2
extension
to C 1—1, 19—1, 19—2
to C anonymous union 19—2
to C class 19—2
to C const 19—2
to C, dangerous 19—12
to C declaration statement 19—2
to C delete 19—2
to C destructor 19—2
to C, implementation dependency 19—12
to C inline function 19—2
to C memberwise assignment 19—2
to C memberwise initialization 19—2
to C multiple inheritance 19—2
to C new 19—2
to C overloading 19—2
to C overloading delete 19—2
to C overloading new 19—2
to C pointer to member 19—2
to C protected 19—2
to C reference type 19—2
to C single precision arithmetic 19—2
to C type checking 19—1
to C user-defined type 19—2
to C, void* pointer type 19—2
to C volatile 19—2
extern
declaration 3—1
linkage of 7—2
linkage specification 7—8
reference declaration 8—13
restriction 7—2
external linkage 3—3

F

F suffix 2—8
f suffix 2—8
field declaration : 9—10
file 2—1, 3—3
scope 3—2
source 2—1
final overrider 10—6
floctal-digit 2—6
float
literal 2—8
type 3—5
type specifier 7—6
<float.h> 2—5
floating
point arithmetic, single precision 4—2
point conversion, implementation dependency 4—2
point conversion, safe 4—2
point integer conversion 4—2
point literal 2—8
point literal, type of 2—8
point type 3—5
point type, implementation dependency 3—5
floating-constant 2—7
floating-suffix 2—8
for
scope of declaration in 6—4
statement 6—3, 6—4
statement, continue in 6—4
statement, declaration in 6—4
form feed 2—7
formal
argument − see also parameter
argument − see parameter
forward
class declaration 9—2, 10—1
declaration 7—3
declaration example, nested class 9—11
fractional-constant 2—7
free
store − see also new, delete
store and constructor anachronism 19—13
store and destructor anachronism 19—13
friend function, scope of 11—6
friend
access specifier and 11—6
class 11—6
class access and 11—6
declaration, overloaded name and 11—5
example 9—2
function, access and 11—5
function example 11—5
function, inline 11—6
function, linkage of 11—6
function, member function and 11—5
function, nested class 9—11
inheritance and 11—6
member function 11—5
scope of 3—2
specifier 7—5
template and 14—5
virtual and 10—7
function

− see also `friend` function, member function, inline function, virtual function
allocation 12—6
argument − see argument
body 8—10
call 5—3
call evaluation, unspecified order of 5—4
call operator 5—3, 13—7
call operator, overloaded 13—8
call, recursive 5—4
call, undefined 5—10
cast, implementation dependency pointer to 5—10
cast, pointer to 5—10
comparison, pointer to 5—13
conversion, pointer to 4—3
declaration 3—1, 8—7
declaration ambiguity 8—11
declaration, ellipsis in 5—4, 8—7
declaration example 8—8
declaration matching, overloaded 13—2
declarator `()` 8—7
definition 8—10
definition 3—2
definition anachronism, C 19—12
definition anachronism, old style 19—12
definition example 8—10
definition, scope of 3—3
linkage specification 7—8
linkage specification overloaded 7—9
member − see member function
member declaration 9—6
name hiding 13—2
name, overloaded 13—1
`operator` 13—6
overloaded − see also overloading
parameter − see parameter
parameter adjustment 8—7
pointer to member 5—11
prototype scope 3—2
return − see `return`
return type − see return type
scope 3—2
scope of friend 11—6
specifier 7—3
template 14—3
template declaration 14—4
template definition 14—4
type 3—6, 8—7
virtual − see virtual function
function-body 8—10
function-definition 8—10
function-like macro 16—4
function-specifier 7—3
fundamental
type 3—5
type conversion − see conversion, user-defined conversion
type, destructor and 12—6

G

generated
constructor − see default constructor
destructor − see default destructor
global
anonymous `union` 9—9
name 3—2
object storage class 3—4
`goto`
initialization and 6—6
statement 6—1, 6—5
grammar 18—1
greater
than operator 5—13
than or equal to operator 5—13

H

handler, exception 15—3
handler 15—1
handler-seq 15—1
handling exception − see exception handling
headers
ANSI C 2—5
library 2—5
standard 2—5
hex number 2—7
hexadecimal literal 2—6
hexadecimal-digit 2—6
hexadecimal-escape-sequence 2—7
hexadecimal-literal 2—5
hiding − see name hiding
horizontal tab 2—7

I

id, qualified 5—2
identifier 2—4, 5—2, 7—1
_, underscore in 2—4
identifier 2—4
identifiers, reserved 17—9
identities and overloading, operator 13—7
id-expression 5—2
id-expression 5—2
`#if` 16—3
`if` statement 6—2, 6—3
`#ifdef` 16—3
`if-else` ambiguity 6—3
`#ifndef` 16—3
implementation
dependency `__STDC__` 16—9
dependency alignment of bit-field 9—10
dependency alignment requirement 3—5
dependency allocation 9—4, 11—2
dependency arithmetic exception 5—1
dependency `asm` 7—8
dependency base class allocation 10—2
dependency bit-field allocation 9—10
dependency change to `const` object 5—10
dependency constructor and `new` 5—8
dependency division 5—11
dependency extension to C 19—12
dependency floating point conversion 4—2
dependency floating point type 3—5
dependency generation of temporary 12—2
dependency integer conversion 4—1
dependency left shift 5—12
dependency linkage of `main()` 3—3
dependency linkage specification 7—8
dependency modulus 5—11
dependency object linkage 7—9
dependency overflow 5—1
dependency parameters to `main()` 3—3
dependency pointer comparison 5—13
dependency pointer integer conversion 5—9
dependency pointer subtraction 5—12
dependency pointer to function cast 5—10
dependency promotion of `wchar_t` 4—1
dependency range of types 2—5
dependency sign of bit-field 9—10

dependency sign of `char` 3—5
dependency `signed unsigned` 4—1
dependency `sizeof` expression 5—7
dependency `sizeof` integral type 3—5
dependency `sizeof` type 3—5
dependency string literal 2—8
dependency type of integer literal 2—6
dependency type of `ptrdiff_t` 5—12
dependency type of `size_t` 5—7
dependency type of `sizeof` expression 2—5
dependency value of `char` literal 2—7
dependency value of multicharacter literal 2—7
dependency `volatile` 7—5
dependency `wchar_t` 3—5
implicit
conversion 4—1, 5—2, 12—3
destructor call 12—5
user-defined conversion 12—4
`#include` 16—3
inclusion
conditional 16—2
source file 16—3
incomplete
type 3—4
type example 3—5
increment
operator 5—5, 5—6, 5—7
operator, overloaded 13—8
indeterminate uninitialized variable 8—11
indirect base class 10—1
indirection 5—6
operator 5—6
inequality operator 5—13
inheritance 10—1
– see also multiple inheritance
and `friend` 11—6
of constructor 12—1
of destructor 12—5
of overloaded operator 13—7
of user-defined conversion 12—4
init-declarator 8—1
init-declarator-list 8—1
initialization 8—10
and `goto` 6—6
and `new` 5—8, 12—9
array 8—11
array of class objects 8—12, 12—9
`auto` 6—6
`auto` object 8—10
base class 12—9, 12—10
character array 8—13
class member 8—11
class object 8—11, 12—8
class object – see also constructor
`const` 7—5, 8—11
`const` member 12—10
`const` pointer 8—11
constructor and 12—8
default 8—11
default constructor and 12—8
definition and 7—2
dynamic 3—4
example, constructor and 12—8
extension to C memberwise 19—2
in block 6—6
jump past 6—3, 6—6
local object 3—4
local `static` 6—6
member 12—9, 12—12
member object 12—10
order of 3—4, 10—3
order of base class 12—10
order of member 12—10
order of virtual base class 12—10
order-of 3—4
overloaded assignment and 12—9
parameter 5—3
pointer to `const` 8—11
pointer to `volatile` 8—11
reference 8—5, 8—13
reference member 12—10
`register` 6—6
run-time 3—4
`static` member 9—9
`static` object 3—4, 8—10, 8—11
`struct` 8—11
`union` 8—12, 9—9
virtual base class 12—10, 12—12
`volatile` pointer 8—11
initializer 8—10
base class 8—10
list `{}` 8—11
member 8—10
scope of member 12—11
temporary and declarator 12—3
initializer 8—10
initializer-clause 8—10
initializer-list 8—10
inline
`friend` function 11—6
function 7—3
function, extension to C 19—2
function, linkage of 3—3, 7—2, 7—3
member function 7—3, 9—8
member function definition 3—3
member function rewriting rules 9—8
`inline` specifier 7—3
`int`
default type 7—5
type 3—5
type specifier 7—6
integer
cast, pointer to 5—9
constant 2—6
conversion 4—1
conversion, `char` 4—1
conversion, floating point 4—2
conversion, implementation dependency 4—1
conversion, implementation dependency pointer 5—9
conversion, `signed unsigned` 4—1
literal 2—6
literal, base of 2—6
literal, implementation dependency type of 2—6
literal, type of 2—6
to pointer cast 5—9
integer-literal 2—5
integer-suffix 2—6
integral
promotion 4—1, 5—4
type 3—5
type, implementation dependency `sizeof` 3—5
value, undefined unrepresentable 4—2
internal linkage 3—3
interpretation
of binary operator 13—7
of unary operator 13—7
invocation, macro 16—5
iteration statement 6—3
iteration-statement 6—3, 6—5
scope 6—4

J

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

K

Kernighan and Ritchie 1—1
keyword 18—1
  anachronism, `overload` 19—12
  list 2—4
Koenig, Andrew 15—1

L

`L`
  prefix 2—7, 2—8
  suffix 2—6, 2—8
`l` suffix 2—6, 2—8
label 6—5
  `case` 6—1, 6—3
  `default` 6—1, 6—3
  name space 6—1
  scope of 3—2, 6—1
  specifier `:` 6—1
labeled statement 6—1
lattice − see DAG, class
layout
  access specifier and object 11—2
  bit-field 9—10
  class object 9—4, 10—3
left
  shift, implementation dependency 5—12
  shift operator 5—12
  shift, undefined 5—12
length of name 2—4
less
  than operator 5—13
  than or equal to operator 5—13
letter definition 17—9
lexical conventions 2—1
library headers 2—5
`<limits.h>` 2—5, 3—5
`#line` 16—8
linkage 3—1, 3—3
  consistency 3—3, 7—2
  consistency example 7—2
  external 3—3
  implementation dependency object 7—9
  internal 3—3
  of class 3—3
  of `const` 3—3, 7—2
  of enumerator 3—3
  of `extern` 7—2
  of `friend` function 11—6
  of inline function 3—3, 7—2, 7—3
  of local name 3—3
  of `main()`, implementation dependency 3—3
  of member function 3—3
  of `static` 3—3, 7—2, 7—9
  of `static` member 3—3
  of template 3—3
  of `typedef` 3—3
  specification 7—8
  specification class 7—8
  specification consistency 7—9
  specification, `extern` 7—8
  specification function 7—8
  specification, implementation dependency 7—8
  specification object 7—9
  specification overloaded function 7—9
  to C 7—8
linkage-specification 7—8
list
  keyword 2—4
  operator 2—4, 13—6
  `{}`, initializer 8—11
literal 2—5, 5—2
  base of integer 2—6
  character 2—7
  decimal 2—6
  `double` 2—8
  `float` 2—8
  floating point 2—8
  hexadecimal 2—6
  implementation dependency string 2—8
  implementation dependency type of integer 2—6
  implementation dependency value of `char` 2—7
  implementation dependency value of multicharacter 2—7
  integer 2—6
  `long` 2—6
  `long double` 2—8
  multicharacter 2—7
  octal 2—6
  type of character 2—7
  type of floating point 2—8
  type of integer 2—6
  `unsigned` 2—6
literal 2—5
local
  class definition 9—11
  class example 9—11
  class member function 9—12
  class, member function in 9—8
  class restriction 9—12
  class restriction, `static` member 9—8
  class, scope of 9—11
  name, linkage of 3—3
  object constructor 3—4
  object destructor 3—4
  object initialization 3—4
  object, `static` 3—4
  object storage class 3—4
  scope 3—2
  `static`, destruction of 6—6
  `static` initialization 6—6
  variable, destruction of 6—5, 6—6
logical
  AND operator 5—14
  AND operator, side effects and 5—14
  OR operator 5—14
  OR operator, side effects and 5—14
  negation operator 5—6, 5—7
`long`
  constant 2—6
  `double` literal 2—8
  `double` type 3—5
  literal 2—6
  type 3—5
  type specifier 7—6
  `typedef` and 7—2
long-suffix 2—6
lvalue 3—7
  assignment and 5—15
  cast 5—10
  conversion 4—1
  modifiable 3—7

M

macro
 definition scope 16—6
 function-like 16—4
 invocation 16—5
 name 16—4
 object-like 16—4
 parameters 16—5
 preprocessor 16—1
 replacement 16—4
 syntax summary 18—11
`main()` 3—3
 implementation dependency linkage of 3—3
 implementation dependency parameters to 3—3
 parameters to 3—3
 `return` from 3—4
management anachronism, memory 19—13
manual organization 1—1
meaning of declarator 8—3
member
 − see also base class member
 access operator, overloaded 13—8
 access ambiguity 10—4
 access, base class 10—1
 access, class 5—4
 access, `protected` 11—7
 access, `struct` default 9—1
 access, `union` default 9—1
 alignment of class 5—10
 array 9—4
 assignment 12—12
 class object 9—4
 constructor order of execution 12—2
 declaration 9—3
 declaration, class 9—3
 declaration, `static` 3—1
 definition 9—7
 definition, `static` 9—9
 destructor order of execution 12—5
 enumerator 7—8
 example, `static` 9—8
 function and access control 12—1
 function and `friend` function 11—5
 function call, constructor and 12—11
 function call, destructor and 12—11
 function call, undefined 9—7
 function, class 9—6
 function, `const` 9—7
 function, constructor and 12—2
 function definition 9—6, 9—7, 9—8
 function definition, inline 3—3
 function, destructor and 12—5
 function example 9—6, 11—5
 function, `friend` 11—5
 function in local class 9—8
 function in nested class 9—8
 function, inline 7—3, 9—8
 function, linkage of 3—3
 function, local class 9—12
 function, nested class 9—10
 function, overloading resolution and 13—3
 function rewriting rules, inline 9—8
 function, `static` 9—6, 9—8
 function template 14—5
 function, `union` 9—9
 function, `volatile` 9—7
 initialization 12—9, 12—12
 initialization, `const` 12—10
 initialization, order of 12—10
 initialization, reference 12—10
 initialization, `static` 9—9
 initializer 8—10
 initializer, scope of 12—11
 linkage of `static` 3—3
 local class restriction, `static` 9—8
 name access 11—1
 name access example 11—3
 name, overloaded 9—4
 object initialization 12—10
 of class type restriction 12—9
 pointer to − see pointer to member
 `static` 9—8
 `static` class 3—4
 storage class, class 3—4
 template and `static` 14—6
 type of `static` 5—6, 9—9
 use, `static` 9—8
member-declaration 9—3
member-declarator 9—3
member-specification 9—3
memberwise
 assignment 13—7
 assignment, extension to C 19—2
 initialization, extension to C 19—2
mem-initializer 12—10
memory
 management − see also `new`, `delete`
 management anachronism 19—13
 model 1—2
missing
 storage class specifier 7—2
 type specifier 7—5
modifiable lvalue 3—7
modulus
 implementation dependency 5—11
 operator 5—11
most derived class 12—10
multicharacter
 literal 2—7
 literal, implementation dependency value of 2—7
multidimensional
 array 8—6
 array declarator 8—6
multiple
 declaration 3—3
 inheritance 10—1, 10—3
 inheritance DAG 10—3
 inheritance, extension to C 19—2
 inheritance, `virtual` and 10—7
multiplication operator 5—11
multiplicative operator 5—11
multiplicative-expression 5—11

N

name 2—4, 3—1, 5—2
 address of qualified 5—6
 and translation unit 3—1
 class − see class name
 declaration 3—1
 elaborated `enum` 7—6
 global 3—2
 hiding 3—2, 3—3, 5—2, 6—5
 hiding, class definition 9—2
 hiding, default parameter and 8—9
 hiding, function 13—2
 hiding, overloading versus 13—2
 hiding, user-defined conversion and 12—5
 length of 2—4
 linkage of local 3—3

macro 16—4
overloaded function 13—1
overloaded member 9—4
point of declaration 3—2
scope of 3—2
space, label 6—1
type − see type name
negation operator, logical 5—6, 5—7
nested
class anachronism, scope of 19—14
class definition 9—10
class definition example 9—10, 9—11
class example 9—10
class forward declaration example 9—11
class `friend` function 9—11
class member function 9—10
class, member function in 9—8
class qualified name 7—6
class, scope of 9—10
type name 9—12
type name example 9—12
type name, scope of 9—12
nested-class-specifier 7—6
`new` 5—7, 5—8, 12—6
array 5—8
array of class objects and 5—8
constructor and 5—8
default constructor and 5—8
extension to C 19—2
extension to C overloading 19—2
implementation dependency constructor and 5—8
initialization and 5—8, 12—9
placement syntax 5—8
scoping and 5—8
storage allocation 5—7
type of 12—7
unspecified order of evaluation 5—8
unspecified value 5—8
new-declarator 5—7
new-expression 5—7
`<new.h>` 2—5
new-initializer 5—7
new-line 2—7
new-placement 5—7
new-type-id 5—7
nondigit 2—4
nonnested class anachronism 19—14
nonvirtual base class DAG 10—3
nonzero-digit 2—6
notation, syntax 1—1
NTCS definition 17—8
null
character `0` 2—8
directive 16—8
pointer 4—2, 4—3, 5—13
pointer conversion 4—3
reference 8—5
statement 6—2
number
hex 2—7
octal 2—7

O

object 1—2, 3—1, 3—7
class − see also class object
complete 1—2
constructor, local 3—4
definition 3—2
destructor and placement of 12—6
destructor, local 3—4
destructor `static` 3—4
initialization, `auto` 8—10
initialization, local 3—4
initialization, `static` 3—4, 8—10, 8—11
layout, access specifier and 11—2
linkage, implementation dependency 7—9
linkage specification 7—9
`static` local 3—4
storage class, global 3—4
storage class, local 3—4
temporary − see temporary
undefined deleted 5—9
unnamed 12—2
object-like macro 16—4
octal
literal 2—6
number 2—7
octal-escape-sequence 2—7
octal-literal 2—5
old
style base class initializer anachronism 19—13
style function definition anachronism 19—12
one-definition rule 3—2
one's complement operator 5—6, 5—7
operand
`const` 5—1
reference 5—1
`volatile` 5—1
operations on class object 9—1
operator
`%=` 5—15
`&=` 5—15
`*=` 5—15
`+=` 5—7, 5—15
`-=` 5—15
`/=` 5—15
`<<=` 5—15
`>>=` 5—15
`^=` 5—15
additive 5—11
address-of 5—6
assignment 5—15, 12—12
bitwise 5—13
bitwise AND 5—13
bitwise exclusive OR 5—14
bitwise inclusive OR 5—14
cast 5—6, 5—9, 8—2
class member access 5—4
comma 5—16
conditional expression 5—14
conversion 5—2, 12—4
decrement 5—5, 5—6, 5—7
default assignment 13—7
`delete` − see `delete`
division 5—11
equality 5—13
example, scope resolution 10—4
function call 5—3, 13—7
function call 13—7
greater than 5—13
greater than or equal to 5—13
identities and overloading 13—7
increment 5—5, 5—6, 5—7
indirection 5—6
inequality 5—13
left shift − see left shift operator
less than 5—13
less than or equal to 5—13
list 2—4, 13—6
logical AND 5—14

logical OR 5—14
logical negation 5—6, 5—7
modulus 5—11
multiplication 5—11
multiplicative 5—11
`new` − see `new`
one's complement 5—6, 5—7
overloaded 5—1
overloading − see also overloaded operator
overloading restrictions 13—7
pointer to member 5—11
precedence of 5—1
relational 5—13
right shift; right shift operator 5—12
scope resolution 3—2, 5—2, 9—7, 10—1, 10—8
shift − see left shift operator, right shift operator
side effects and comma 5—16
side effects and logical AND 5—14
side effects and logical OR 5—14
`sizeof` 5—6, 5—7
subscripting 5—3, 13—7
unary 5—6
unary minus 5—6, 5—7
unary plus 5—6
use, scope resolution 9—8
`|=` 5—15
`operator`
function 13—6
overloaded 13—6
operator 13—6
operator-function-id 13—6
optimization of temporary − see elimination of temporary
OR
operator, bitwise exclusive 5—14
operator, bitwise inclusive 5—14
operator, logical 5—14
operator, side effects and logical 5—14
order
of argument evaluation 5—4
of argument evaluation, unspecified 5—4
of base class initialization 12—10
of destruction of temporary 12—3
of evaluation `new`, unspecified 5—8
of evaluation of expression 5—1
of evaluation, unspecified 5—1
of execution, base class constructor 12—2
of execution, base class destructor 12—5
of execution, constructor and array 12—2
of execution, constructor and `static` objects 12—9
of execution, destructor 12—5
of execution, destructor and array 12—5
of execution, destructor and `static` objects 12—9
of execution, member constructor 12—2
of execution, member destructor 12—5
of function call evaluation, unspecified 5—4
of initialization 3—4, 10—3
of member initialization 12—10
of virtual base class initialization 12—10
order-of
construction 3—4
destruction 3—4
initialization 3—4
organization, manual 1—1
out of range value, undefined conversion 4—2
overflow 5—1
implementation dependency 5—1
`overload` keyword anachronism 19—12
overloaded
assignment and initialization 12—9
assignment operator 13—7
binary operator 13—7
decrement operator 13—8
function, address of 5—6, 13—6
function ambiguity detection 13—3
function and standard conversion 13—4
function call operator 13—8
function call resolution − see also argument matching, overloading resolution
function declaration matching 13—2
function, linkage specification 7—9
function name 13—1
increment operator 13—8
member access operator 13—8
member name 9—4
name and access declaration 11—5
name and `friend` declaration 11—5
`operator` 13—6
operator 5—1
operator 13—6
operator and default parameter 13—7
operator, inheritance of 13—7
subscripting operator 13—8
unary operator 13—7
overloading 8—7, 9—2, 13—1
and access 13—3
and `const` 13—1
and default parameter 8—10
and `delete` 12—7
and derived class 13—2
and `enum` 13—2
and pointer 13—1
and pointer versus array 13—2
and reference 13—1
and return type 13—1
and scope 13—2
and `static` 13—1
and `typedef` 13—2
and `volatile` 13—1
`delete`, extension to C 19—2
example 13—1
extension to C 19—2
`new`, extension to C 19—2
operator identities and 13—7
postfix `++` and `--` 13—8
prefix `++` and `--` 13—8
resolution 13—3
resolution and access control 10—4
resolution and conversion 13—3
resolution and default argument 13—3
resolution and ellipsis 13—3, 13—4
resolution and member function 13—3
resolution and pointer conversion 13—6
resolution and promotion 13—4
resolution and standard conversion 13—4
resolution and user-defined conversion 13—4, 13—5
resolution exact match 13—4
resolution rules 13—4
resolution, template function 14—3
resolution trivial conversions 13—4
restriction 13—7
subsequence rule 13—3
versus name hiding 13—2
overrider, final 10—6

P

parameter 1—3, 8—10
adjustment, array 8—7
adjustment, function 8—7
declaration 8—7

default − see default parameter
example, unnamed 8—10
initialization 5—3
list example, variable 8—8
list, variable 5—4, 8—7
reference 8—4
scope of 3—2
temporary and default 12—3
`void` 8—7
parameter-declaration 8—7
parameterized type − see template
parameters
macro 16—5
to `main()` 3—3
to `main()`, implementation dependency 3—3
parentheses
and ambiguity 5—8
in declaration 8—3
parenthesized expression 5—2
phases, translation 2—1
placement
of object, destructor and 12—6
syntax, `new` 5—8
pm-expression 5—11
point
of declaration class name 9—3
of declaration enumerator 3—2
of declaration name 3—2
of definition, enumerator 7—7
pointer
− see also `void*`
arithmetic 5—11, 5—12
assignment, `const` 5—15
assignment to 5—15
assignment, `volatile` 5—15
cast, integer to 5—9
comparison 5—13
comparison, implementation dependency 5—13
comparison, undefined 5—12, 5—13
comparison, `void*` 5—13
conversion 4—2
conversion ambiguity 4—3
conversion, array 4—3
conversion, base class 4—3
conversion, `const void*` 4—3
conversion, `const volatile void*` 4—3
conversion, derived class 4—3
conversion, null 4—3
conversion, overloading resolution and 13—6
conversion, `void*` 4—3
conversion, `volatile const void*` 4—3
conversion, `volatile void*` 4—3
conversion, zero 4—3
conversions, cv-qualifier 4—2
declaration 8—4
declarator `*` 8—4
example, constant 8—4
initialization, `const` 8—11
initialization, `volatile` 8—11
integer conversion, implementation dependency 5—9
null 4—2, 4—3, 5—13
overloading and 13—1
subtraction, implementation dependency 5—12
terminology 3—6
to abstract class 10—9
to bit-field restriction 9—10
to `const` assignment 5—15
to `const` initialization 8—11
to function cast 5—10
to function cast, implementation dependency 5—10
to function comparison 5—13
to function conversion 4—3
to integer cast 5—9
to member 3—6, 5—11
to member anachronism, cast of 19—13
to member, assignment 5—15
to member, assignment to 5—15
to member cast 5—10
to member constant expression 5—6
to member conversion 4—3
to member conversion ambiguity 4—3
to member conversion anachronism 19—13
to member declarator `::*` 8—5
to member example 8—5
to member, extension to C 19—2
to member function 5—11
to member function, undefined bound 19—13
to member operator 5—11
to member `void*` conversion 4—4
to member, zero assignment to 5—15
to pointer cast 5—10
to `volatile` assignment 5—15
to `volatile` casting 5—10
to `volatile` initialization 8—11
type 3—6
type extension to C, `void*` 19—2
versus array, overloading and 13—2
zero 4—2, 4—3, 5—13
zero assignment to 5—15
polymorphic
class 10—6
type 10—6
postfix
`++` and `--` 5—5
`++` and `--`, overloading 13—8
expression 5—3
pragma directive 16—8
`#pragma` 16—8
precedence of operator 5—1
prefix
`++` and `--` 5—7
`++` and `--`, overloading 13—8
`L` 2—7, 2—8
preprocessing 16—1
directive 16—1
syntax summary 18—11
preprocessing-token 2—2
preprocessor, macro 16—1
primary expression 5—2
`private` 11—1
base class 11—2
program 3—3
environment 3—3
start 3—3, 3—4
termination 3—3, 3—4
termination and destructor 12—5
promotion
integral 4—1, 5—4
of `wchar_t`, implementation dependency 4—1
overloading resolution and 13—4
`protected` 11—1
extension to C 19—2
member access 11—7
protection − see access control
`ptrdiff_t` 5—12
implementation dependency type of 5—12
ptr-operator 8—2
`public` 11—1
base class 11—2
punctuators 2—4
pure
specifier 9—3

virtual destructor 12—5
virtual function 10—8, 10—9
virtual function call, undefined 10—9, 12—11
virtual function definition 10—8
virtual function example 10—8
pure-specifier 9—3

Q

qualified
id 5—2
name, address of 5—6
name, nested class 7—6
qualified-class-specifier 7—6
qualified-id 5—2
qualified-type-specifier 7—6
question mark 2—7
quote
double 2—7
single 2—7

R

range of types, implementation dependency 2—5
readonly memory, undefined 7—5
recursive function call 5—4
redefinition
enumerator 7—7
`typedef` 7—4
reference 3—6
and argument passing 8—13
and `return` 8—13
and temporary 8—13
argument 5—3
assignment 8—13
assignment to 5—15
call by 5—3
cast 5—10
`const` 8—13
conversion 4—3
conversion ambiguity 4—3
conversions, cv-qualifier 4—3
declaration 8—4
declaration, `extern` 8—13
declarator `&` 8—4
expression 5—2
initialization 8—5, 8—13
member initialization 12—10
null 8—5
operand 5—1
overloading and 13—1
parameter 8—4
restriction 8—5
`sizeof` 5—7
temporary, scope of 8—14
to base class 4—3
type, extension to C 19—2
`volatile` 8—14
`register`
declaration 7—2
initialization 6—6
restriction 7—2
relational operator 5—13
relational-expression 5—13
remainder operator − see modulus operator
replacement, macro 16—4
reraise 15—2
rescanning and replacement 16—6
reserved
identifier 2—4
identifiers 17—9
word − see keyword
resolution
overloading − see overloading resolution
scoping ambiguity 10—4
template function overloading 14—3
restriction
address of bit-field 9—10
alignment 5—10
anonymous `union` 9—9
`auto` 7—2
bit-field 9—10
constructor 12—1, 12—2
default assignment operator 12—12
default copy constructor 12—12
destructor 12—5
enumerator 7—7
`extern` 7—2
local class 9—12
member of class type 12—9
overloading 13—7
pointer to bit-field 9—10
reference 8—5
`register` 7—2
`static` 7—2
`static` member local class 9—8
`union` 9—9, 12—2
restrictions, operator overloading 13—7
rethrow 15—2
return
type 8—7
type, class object 12—9
type conversion 6—5
type, overloading and 13—1
`return` 6—5
constructor and 6—5
from `main()` 3—4
reference and 8—13
statement − see also `return`
rewriting rules, inline member function 9—8
right shift operator 5—12
Ritchie, Kernighan and 1—1
ROMable type 7—5
rounding 4—2
rule, one-definition 3—2
rules
for *condition*s 6—2
inline member function rewriting 9—8
overloading resolution 13—4
summary, scope 10—9
type conversion 4—2
run-time initialization 3—4

S

safe floating point conversion 4—2
s-char 2—8
s-char-sequence 2—8
scope 3—1
class 3—2
destructor and exit from 6—5
difference from C 19—1
file 3—2
function 3—2
function prototype 3—2
iteration-statement 6—4
local 3—2
macro definition 16—6
of class definition 9—2
of class name 9—2

of declaration in `for` 6—4
of default parameter 8—9
of `delete` example 12—8
of enumerator class 7—8
of `friend` 3—2
of friend function 11—6
of function definition 3—3
of label 3—2, 6—1
of local class 9—11
of member initializer 12—11
of name 3—2
of nested class 9—10
of nested class anachronism 19—14
of nested type name 9—12
of parameter 3—2
of reference temporary 8—14
overloading and 13—2
resolution operator 3—2, 5—2, 9—7, 10—1, 10—8
resolution operator example 10—4
resolution operator use 9—8
rules summary 10—9
selection-statement 6—2
scoping
ambiguity resolution 10—4
and `new` 5—8
selection statement 6—2
selection-statement 6—2
scope 6—2
semantics, class member 5—4
separate
compilation 2—1
translation 2—1
sequence, statement 6—1
sequencing operator − see comma operator
`setterminate()` 15—5
`setunexpected()` 15—5
shift operator − see left shift operator, right shift operator
shift-expression 5—12
`short`
type 3—5
type specifier 7—6
`typedef` and 7—2
side
effects 5—1
effects and comma operator 5—16
effects and logical AND operator 5—14
effects and logical OR operator 5—14
sign
of bit-field, implementation dependency 9—10
of `char`, implementation dependency 3—5
sign 2—8
`signed`
`char` type 3—5
character 3—5
`typedef` and 7—2
`unsigned`, implementation dependency 4—1
`unsigned` integer conversion 4—1
simple-escape-sequence 2—7
simple-type-specifier 7—5
single
precision arithmetic, extension to C 19—2
precision floating point arithmetic 4—2
quote 2—7
`sizeof`
array 5—7
class object 5—7
difference from C 19—1
empty class 9—1
expression, implementation dependency 5—7
expression, implementation dependency type of 2—5
integral type, implementation dependency 3—5
operator 5—6, 5—7
reference 5—7
string 2—8
type, implementation dependency 3—5
`size_t` 5—7
implementation dependency type of 5—7
source
file 2—1
file inclusion 16—3
space, white 2—3
special member function − see also constructor, destructor, inline function, user-defined conversion, virtual function
specifier
access − see access specifier
`auto` 7—2
declaration 7—1
`friend` 7—5
function 7—3
`inline` 7—3
missing storage class 7—2
`static` 7—2
storage class 7—2
`template` 7—5
type − see type specifier
`typedef` 7—4
`virtual` 7—4
stack unwinding 15—3
standard
conversion 4—1
conversion, overloaded function and 13—4
conversion, overloading resolution and 13—4
headers 2—5
start, program 3—3, 3—4
statement 6—1
− see also `return`, `return`
`break` 6—5
compound 6—2
`continue` 6—5
`continue` in `for` 6—4
declaration 6—5
declaration in `for` 6—4
declaration in `switch` 6—3
`do` 6—3, 6—4
empty 6—2
expression 6—2
extension to C declaration 19—2
`for` 6—3, 6—4
`goto` 6—1, 6—5
`if` 6—2, 6—3
iteration 6—3
jump 6—5
labeled 6—1
null 6—2
selection 6—2
sequence 6—1
`switch` 6—2, 6—3, 6—5
syntax summary 18—10
`while` 6—3, 6—4
`{}`, block 6—2
statement 6—1
static type 1—3
`static`
class member 3—4
destruction of local 6—6
initialization, local 6—6
linkage of 3—3, 7—2, 7—9
local object 3—4
member 9—8
member declaration 3—1

member definition 9—9
member example 9—8
member function 9—6, 9—8
member initialization 9—9
member, linkage of 3—3
member local class restriction 9—8
member, template and 14—6
member, type of 5—6, 9—9
member use 9—8
object, destructor 3—4
object initialization 3—4, 8—10, 8—11
objects order of execution, constructor and 12—9
objects order of execution, destructor and 12—9
overloading and 13—1
restriction 7—2
specifier 7—2
storage class 3—4
variable, template and 14—6
<stdarg.h> 8—7
__STDC__ 16—9
implementation dependency 16—9
<stddef.h> 2—5, 2—7, 2—8, 5—7
<stdlib.h> 2—5, 3—4
storage
allocation new 5—7
class 3—1, 3—4
class, auto 3—4
class, class member 3—4
class declaration 7—2
class, global object 3—4
class, local object 3—4
class specifier 7—2
class specifier, missing 7—2
class, static 3—4
management – see new, delete
of array 8—7
string
concatenation 2—8
constant 2—8
definition 17—8
distinct 2—8
literal 2—8
literal concatenation, undefined 2—8
literal, implementation dependency 2—8
literal, type of 2—8
literal, undefined change to 2—8
sizeof 2—8
terminator 0 2—8
type of 2—8
wide-character 2—8
string-literal 2—8
struct 3—6
class versus 9—1
default member access 9—1
initialization 8—11
type specifier 7—6
structure 3—6, 9—1
tag – see class name
sub-object 1—2
subscripting
example 8—6
explanation 8—6
operator 5—3, 13—7
operator, overloaded 13—8
subsequence rule, overloading 13—3
subtraction
implementation dependency pointer 5—12
operator 5—11
suffix
E 2—8
F 2—8
L 2—6, 2—8
U 2—6
f 2—8
l 2—6, 2—8
u 2—6
summary
class declaration syntax 18—9
compatibility with ANSI C 19—2
compatibility with C 19—1
declaration syntax 18—6
declarator syntax 18—8
exception handling syntax 18—12
expression syntax 18—1
macro syntax 18—11
preprocessing syntax 18—11
scope rules 10—9
statement syntax 18—10
syntax 18—1
template syntax 18—12
switch
statement 6—2, 6—3, 6—5
statement, declaration in 6—3
synonym, type name as 7—4
syntax
class member 5—4
notation 1—1
summary 18—1
summary, class declaration 18—9
summary, declaration 18—6
summary, declarator 18—8
summary, exception handling 18—12
summary, expression 18—1
summary, macro 18—11
summary, preprocessing 18—11
summary, statement 18—10
summary, template 18—12

T

template 14—1
and friend 14—5
and static member 14—6
and static variable 14—6
class 14—2
class declaration 14—4
class definition 14—4
declaration 14—4
function 14—3
function declaration 14—4
function definition 14—4
function overloading resolution 14—3
linkage of 3—3
member function 14—5
syntax summary 18—12
type equivalence 14—3
template 14—1
specifier 7—5
template-argument 14—2
template-argument-list 14—2
template-class-id 14—2
template-declaration 14—1
template-name 14—2
template-parameter 14—1
template-parameter-list 14—1
temporary 12—2
and declarator initializer 12—3
and default parameter 12—3
constructor for 12—2
destruction of 12—3
destructor for 12—2

elimination of 12—2
implementation dependency generation of 12—2
order of destruction of 12—3
reference and 8—13
scope of reference 8—14
terminate() 15—4, 15—5
termination
and destructor, program 12—5
program 3—3, 3—4
terminator 0, string 2—8
terminology, pointer 3—6
this 5—2
anachronism, assignment to 19—13
and constructor anachronism 19—13
and destructor anachronism 19—13
pointer − see this
type of 9—7
throw 15—1
throw-expression in conditional-expression 5—14
throw-expression 15—1
throwing, exception 15—2
throw-point 15—1
token 2—3, 2—5
token 2—3
translation
phases 2—1
separate 2—1
unit 2—2
unit, name and 3—1
translation@unit 2—1
trigraph 2—1, 2—2
trivial conversions, overloading resolution 13—4
truncation 4—2
try 15—1
try-block 15—1
type 3—1
ROMable 7—5
ambiguity, declaration 7—1
arithmetic 3—5
array 3—6
char 3—5
checking, argument 5—3
checking, default parameter 8—9
checking, extension to C 19—1
class and 9—1
const 7—5
conversion − see also conversion
conversion, argument 12—3
conversion, explicit − see casting
conversion rules 4—2
declaration 8—3
declaration consistency 3—3
declaration, typedef as 7—4
definition, class name as 9—2
derived 3—6
double 3—5
dynamic 1—3
enumerated 3—5
enumeration underlying 7—7
equivalence 7—4, 9—2
equivalence, template 14—3
example, incomplete 3—5
extension to C reference 19—2
extension to C user-defined 19—2
float 3—5
function 3—6, 8—7
fundamental 3—5
generator − see template
implementation dependency sizeof 3—5
incomplete 3—4
int 3—5
int, default 7—5
long 3—5
long double 3—5
name 3—7, 8—2
name as synonym 7—4
name example 8—2
name example, nested 9—12
name, nested 9—12
name, scope of nested 9—12
of bit-field 9—10
of character literal 2—7
of constructor 12—2
of conversion 12—4
of delete 12—7
of enum 7—6, 7—7
of floating point literal 2—8
of integer literal 2—6
of integer literal, implementation dependency 2—6
of new 12—7
of ptrdiff_t, implementation dependency 5—12
of size_t, implementation dependency 5—7
of sizeof expression, implementation dependency 2—5
of static member 5—6, 9—9
of string 2—8
of string literal 2—8
of this 9—7
pointer 3—6
polymorphic 10—6
short 3—5
signed char 3—5
specifier, char 7—6
specifier, class 7—6
specifier, double 7—6
specifier, enum 7—6
specifier, float 7—6
specifier, int 7—6
specifier, long 7—6
specifier, missing 7—5
specifier, short 7—6
specifier, struct 7—6
specifier, union 7—6
specifier, unsigned 7—6
specifier, void 7—6
specifier, volatile 7—5
static 1—3
unsigned 3—5
unsigned char 3—5
void 3—6
void* 3—6
volatile 7—5
wchar_t 3—5
wchar_t underlying 3—5
typedef 3—7
and long 7—2
and short 7—2
and signed 7—2
and unsigned 7—2
as type declaration 7—4
class name 7—4, 7—5, 9—3
declaration 3—1
example 7—4
linkage of 3—3
overloading and 13—2
redefinition 7—4
specifier 7—4
typedef-name 7—4
typeid 5—6
type-id 8—2
type-id-list 15—4
Typeinfo 17—5

type-parameter 14—1
types
  floating − see floating point type
  implementation dependency range of 2—5
  integral − see integral type
type-specifier 7—5

U

`U` suffix 2—6
`u` suffix 2—6
unary
  expression 5—6
  minus operator 5—6, 5—7
  operator 5—6
  operator, interpretation of 13—7
  operator, overloaded 13—7
  plus operator 5—6
unary-expression 5—6
unary-operator 5—6
`#undef` 16—6
undefined
  bound pointer to member function 19—13
  change to `const` object 7—5
  change to string literal 2—8
  class, cast to 5—10
  conversion out of range value 4—2
  `delete` 5—9
  deleted object 5—9
  division by zero 5—1, 5—11
  escape sequence 2—7
  function call 5—10
  left shift 5—12
  member function call 9—7
  pointer comparison 5—12, 5—13
  pure virtual function call 10—9, 12—11
  readonly memory 7—5
  string literal concatenation 2—8
  unrepresentable integral value 4—2
  value `delete` 5—9
underlying
  type, enumeration 7—7
  type, `wchar_t` 3—5
underscore
  character _ 2—4
  in identifier _ 2—4
`unexpected()` 15—5
uninitialized variable, indeterminate 8—11
`union` 3—6, 9—9
  access control, anonymous 9—9
  anonymous 9—9
  `class` versus 9—1
  constructor 9—9
  default member access 9—1
  destructor 9—9
  extension to C anonymous 19—2
  global anonymous 9—9
  initialization 8—12, 9—9
  member function 9—9
  restriction 9—9, 12—2
  restriction, anonymous 9—9
  type specifier 7—6
unknown argument type 8—7
unnamed
  bit-field 9—10
  class 7—5
  object 12—2
  parameter example 8—10
unrepresentable integral value, undefined 4—2
`unsigned`
  arithmetic 3—5
  `char` type 3—5
  constant 2—6
  implementation dependency `signed` 4—1
  integer conversion, `signed` 4—1
  literal 2—6
  type 3—5
  type specifier 7—6
  `typedef` and 7—2
unsigned-suffix 2—6
unspecified
  argument to constructor 5—8
  destructor call 6—6
  expression 5—4
  order of argument evaluation 5—4
  order of evaluation 5—1
  order of evaluation `new` 5—8
  order of function call evaluation 5—4
  value `new` 5—8
unwinding, stack 15—3
user-defined
  conversion 5—2, 12—3, 12—4
  conversion and name hiding 12—5
  conversion, implicit 12—4
  conversion, inheritance of 12—4
  conversion, overloading resolution and 13—4, 13—5
  conversion, virtual 12—4
  type, extension to C 19—2

V

value
  call by 5—3
  `delete`, undefined 5—9
  `new`, unspecified 5—8
  of `char` literal, implementation dependency 2—7
  of enumerator 7—7
  of multicharacter literal, implementation dependency 2—7
  undefined conversion out of range 4—2
  undefined unrepresentable integral 4—2
variable
  argument list 8—7
  indeterminate uninitialized 8—11
  parameter list 5—4, 8—7
  parameter list example 8—8
  template and `static` 14—6
vertical tab 2—7
virtual
  base class 10—3
  base class DAG 10—3
  base class dominance 10—5
  base class initialization 12—10, 12—12
  base class initialization, order of 12—10
  destructor 12—5
  destructor, pure 12—5
  function 10—6
  function access 11—7
  function call 10—8
  function call, constructor and 12—11
  function call, destructor and 12—11
  function call, undefined pure 10—9, 12—11
  function definition 10—7
  function definition, pure 10—8
  function example 10—6, 10—7
  function example, pure 10—8
  function, pure 10—8, 10—9
  user-defined conversion 12—4
`virtual`
  and `friend` 10—7

and multiple inheritance 10—7
specifier 7—4
`void`
parameter 8—7
type 3—6
type specifier 7—6
`void&` 8—4
`void*`
conversion, pointer to member 4—4
pointer comparison 5—13
pointer conversion 4—3
pointer type extension to C 19—2
type 3—6
`volatile` 3—6
assignment, pointer to 5—15
casting, pointer to 5—10
`const void*` pointer conversion 4—3
constructor and 9—8, 12—1
destructor and 9—8, 12—5
extension to C 19—2
implementation dependency 7—5
initialization, pointer to 8—11
member function 9—7
operand 5—1
overloading and 13—1
pointer assignment 5—15
pointer initialization 8—11
reference 8—14
type 7—5
type specifier 7—5
`void*` pointer conversion 4—3

W

`wchar_t` 2—7, 2—8
implementation dependency 3—5
implementation dependency promotion of 4—1
type 3—5
underlying type 3—5
`while` statement 6—3, 6—4
white
space 2—3
space 2—3
wide-character 2—7
string 2—8

X

`X(X&)` – see copy constructor

Z

zero
assignment to pointer 5—15
assignment to pointer to member 5—15
pointer 4—2, 4—3, 5—13
pointer conversion 4—3
undefined division by 5—1, 5—11
width of bit-field 9—10