

## Index

$\&$  (operator), 66, 141, 142  
< >, 10  
-> (arrow operator), 66, 124, 146, 148  
\* (operator), 66, 141-142  
: (used in inheritance), 60  
:: (scope resolution operator), 22, 112-113, 438  
. (dot operator), 23, 25-26, 146, 148  
<< (output operator), 13-14, 292, 338  
>> (input operator), 13-14, 299, 338  
[] (array subscripting operator), 222-226  
~ (for destructor function), 44

## A

abort() function, 378  
adjustfield format flag, 275, 276  
<algorithm> header, 509  
Algorithms, 476, 509-518  
  table of, 510-512  
allocator type, 477  
Allocators, 477  
Anonymous unions, 69-70, 74-75  
append() function, 522-523  
argc, argv convention, 3  
Arguments, default. *See* Default arguments  
Array(s)  
  based I/O, 466-470  
  bounds checking, 151-153, 169-171, 225-226  
  dynamic, 480  
  dynamically allocated, 134, 136-137, 165-166  
  of objects, 119-123, 137-138  
Arrow operator (->), 66, 124, 146, 148  
asm statement, 463, 464, 465  
Assembly language instructions, embedding,  
  463, 464, 465  
assign(), 522

Assignment operations  
  and copy constructors, 168  
  and functions, 149-150  
  and objects, 89-94  
  and overloaded assignment operator,  
    218-221

## B

bad() function, 335  
bad\_alloc exception, 401  
badbit flag, 335  
bad\_typeid exception, 410  
Base class  
  access control, 234-238, 240-243  
  definition of, 59  
  indirect, 252  
  inheriting, general form for, 61-62  
  passing arguments from derived class to,  
    244-245, 246-249  
  virtual, 259-261  
basefield format flag, 275, 276  
basic\_fstream, 275  
basic\_ifstream, 275  
basic\_ios, 274  
basic\_iostream, 274, 275  
basic\_istream, 274, 275  
basic\_ofstream, 275  
basic\_ostream, 274, 275  
basic\_streambuf, 274  
basic\_string class, 519  
before() function, 409  
begin() function, 482, 487, 525  
Binary I/O  
  and character translations, 314, 321-322  
  unformatted, 320-326

Binary operators, overloading, 199-211  
 bool data type, 30, 32, 207  
 boolalpha  
   format flag, 275, 276, 290-291  
   I/O manipulator, 288, 290-291

**C**

C++  
   differences between C and, 28-32, 532  
   similarities between C and, 2, 3  
 C++, Standard, 2, 7-12  
   differences between C++ and, 8-9  
 Call-by-reference parameter passing, 141-144  
 Casts in C++, 420  
 catch statement, 386-388, 389-390, 392-393  
 catch(...), 394, 395-397  
   as default catch statement, 396-397  
 cerr stream, 273  
 cin stream, 14, 273  
 class keyword, 68-69  
 Classes, 21-28  
   abstract, 358  
   base. *See* Base class  
   declaration, general form of, 21, 240  
   derived. *See* Derived class  
   forward reference to, 110-111  
   generic. *See* Generic classes  
   members of. *See* Members, class  
   polymorphic, 350  
   relationship to structures and unions, 68-69  
   template. *See* Generic classes  
 clear() function, 335  
 clock() function, 51  
 clog stream, 273  
 close() function, 315  
 <cmath> header, 10  
 Comments, 19-20  
 Compiler, 3  
   working with old, 12, 15, 308  
 compose() function, 524  
 Console I/O, 13-18, 270, 338  
 const member functions, 455-458

const\_cast, 429, 430-431  
 Constructors, 43-44, 45, 459-462  
   as in-line functions, 81-82  
   copy. *See* Copy constructors  
   and default arguments, 181  
   example uses for, 46-51  
   and inheritance, 244-249  
   initializing array of objects with, 120-121,  
   122-123  
   and multiple inheritance, 252-253, 256-257  
   overloading, 161-166  
   parameters and, 52-58  
   and passing objects to functions, 99  
   variable declarations in, 44  
 Containers, 467, 476  
   table of, 479  
 Conversion functions, 446-449  
 Copy constructors, 101, 106, 167-174, 221  
   and default arguments, 183  
   general form of, 169  
 count() algorithm, 510, 513-514  
 count\_if() algorithm, 510, 513-514  
 cout stream, 13, 273  
 <cstdlib> header, 132  
 c\_str() function, 524-525  
 <cstring> header, 10

**D**

Data type, object type as, 5  
 dec  
   format flag, 275, 276, 279  
   I/O manipulator, 288  
 Decrement operator (--), overloading for  
   postfix and prefix, 210-211, 217-218  
 Default arguments, 177-183, 199  
   and ambiguity, 188  
 delete operator, 130-133  
   and dynamically allocated arrays, 134  
 Derived class  
   definition of, 59  
   general form of, 61-62  
   inheriting multiple base classes, 252-257

passing arguments to base class from,  
     244-245, 246-249  
 pointers to, 347-349  
 and virtual base classes, 259-261  
**Destructors**, 44-45  
     as in-line functions, 81  
     example uses for, 48-51  
     and inheritance, 244-249  
     and multiple inheritance, 252, 256-257  
     and parameters, 53  
     and passing objects to functions, 99-101  
     and returning objects from functions,  
         104-106  
**Dot operator (.)**, 23, 25-26, 146, 148  
**dynamic\_cast**, 420-428  
     to replace typeid, 422, 425-426  
  
**E**  
 early binding, 362-363  
**Encapsulation**, 4, 7, 61, 69, 450  
**end( )** function, 482, 487, 525  
**endl** I/O manipulator, 288  
**ends** I/O manipulator, 468  
**eof( )** function, 315-316, 318, 335, 467  
**eofbit** flag, 335  
**erase( )**, function, 482, 484, 487, 523, 527-528  
**Error handling**, run time. *See* Exception  
     handling  
**Exception handling**, 130, 131, 372, 386-400  
     and catching all exceptions, 394, 395-397  
     general operation of, 386-389  
     and new, 401-404  
     and restricting exceptions thrown, 394-395,  
         397-399  
     and rethrowing exceptions, 395, 399-400  
**exit( )** function, 389  
**explicit** specifier, 460, 462  
**Extraction operator (>>)**. *See* Input operator  
**Extractors** (extractor functions), creating,  
     299-302

**F**

**fail( )** function, 335  
**failbit** flag, 335  
**false** value, 30, 207  
**File I/O**  
     basics, 313-320  
     and character translations, 314  
     customized, 338  
     and console I/O, 270, 338  
**File pointers**  
     get, 332, 333-334  
     put, 332-333  
**fill( )** function, 283-285  
**find( )** function, 504, 507, 524  
**fixed**  
     format flag, 275, 276  
     I/O manipulator, 288  
**flags( )** function, 278, 280, 282  
**floatfield** format flag, 275, 276  
**flush( )** function, 328-329  
**flush** I/O manipulator, 288  
**fmtflags**, 275, 276  
**Format flags**, I/O, 275-283  
**Forward reference**, 110-111  
**fprintf( )** function, 316  
**free( )** function, 49, 130, 131, 132  
**Friend functions**, 107-113, 126  
**friend** keyword, 107  
**fscanf( )** function, 316  
**fstream** class, 275, 313, 319  
**<fstream>** header, 313  
**Function**, finding address of overloaded,  
     189-191  
**Function overloading**, 5, 33-37  
     and ambiguity, 187-188  
     constructor, 161-166  
     and default arguments, 177-181, 188  
     vs. generic functions, 377, 379  
     and in-lining, 78-79  
     vs. virtual functions, 351-352  
     *See also* Copy constructors  
**<functional>** header, 478

**Function(s)**

- conversion, 446-449
- friend, 107-113
- generated, 375
- generic. *See* Generic functions
- in assignment statements, using, 149-150
- in-line. *See* In-line functions
- member. *See* Member functions
- objects, 478
- operator. *See* Operator functions
- parameterless, 28-29
- passing objects to, 96-101
- passing references to, 140-145
- pointers to overloaded, 189-191
- predicate, 477-478
- prototyping, 29, 30-31
- return value and, 29
- returning objects from, 102-106
- returning references from, 149-153
- virtual. *See* Virtual functions

**G**

- gcount( ) function, 321
- Generated function, 375
- Generic class(es), 274, 372, 380-386
  - declaration, general form of, 380
  - with multiple generic data types, 385-386
- Generic functions, 372, 373-379, 380-386
  - explicitly overloading, 378-379
  - general form for, 373
  - with multiple generic types, 376-377
  - versus overloaded functions, 377, 379
- get( ) function, 321, 322, 323, 327, 328
- Get pointer, 332, 333-334
- getc( ) function, 327
- getline( ) function, 327-328, 329
- good( ) function, 335, 337
- goodbit flag, 335

**H**

- Headers, 8-11, 444
- hex
  - format flag, 275, 276, 279

- I/O manipulator, 288
- Hierarchical classification, 6, 7, 59

**I**

- ifstream class, 275, 313
- #include statement, 9-10, 11
- Increment operator ( + + ), overloading for postfix and prefix, 210-211, 217-218
- Inheritance, 4, 6, 7, 59-65
  - and class access control, 234-243
  - and constructors and destructors, 244-249
  - and friend functions, 109
  - general form for class, 234
  - multiple, 252-257
  - and virtual base classes, 259-261
- Initialization of objects, 43-44, 46-47, 134
  - copy constructors and, 168-171
  - overloaded constructors and, 161-166
- In-line functions, 75-79
  - automatic, 80-82
    - to define constructors and destructors, 81-82
    - versus parameterized macros, 77
  - inline specifier, 76
- Input operator ( > > ), 13-14, 299, 338
- insert( ) function, 482, 484, 487-488, 491, 492, 523, 527-528
- Inserters (inserter functions), creating, 292-297
- Insertion operator ( < < ). *See* Output operator
- int, default to, 29
- internal
  - format flag, 275, 276
  - I/O manipulator, 288
- I/O
  - array-based, 466-470
  - binary. *See* Binary I/O
  - console, 13-18, 270, 338
  - customized, and files, 338-340
  - file. *See* File I/O
  - formatted, 275-291
  - inserters and extractors, 292-302
  - library, 270



manipulators. *See* Manipulators, I/O operators, 13  
 random access, 331-334  
 status, checking, 334-337  
 streams, 273-274, 313

I/O library, 270  
 C++ vs. Standard C++, 319-320  
 <iomanip> header, 287  
 ios class, 274, 275, 276, 277, 313  
   format flags, 275-278  
   I/O status flags, 334-335  
 ios::app, 314  
 ios::ate, 314  
 ios::beg, 331  
 ios::binary, 314, 321  
 ios::cur, 331  
 ios::end, 331  
 ios::in, 314, 319  
 ios::nocreate, 319-320  
 ios::noreplace, 320  
 ios::out, 314, 319  
 ios::trunc, 314  
 iostate type, 334-335  
 ostream class, 275  
 <ostream> header, 9, 14, 274, 275  
 ostream.h header file, 9  
 is\_open() function, 315  
 istream class, 275, 300, 313  
 istrstream class, 466, 467  
 iterator type, 477, 487  
 iterators, 476-477, 486-487

**K**

Key/value pairs in maps, 476, 502, 505  
 Keywords, C++, table of, 39

**L**

Late binding, 362-363  
 left  
   format flag, 275, 276  
   I/O manipulator, 288

Libraries and namespaces, 437, 440  
 Linkage specifiers, 463-465  
 list class, 476, 490-501  
   member functions, table of, 492-494  
 Local variables, declaring, 29-30, 31  
 Logical operators, overloading, 207-208

**M**

Macros, parameterized, 77  
 main(), 3, 30, 532  
 make\_pair(), 505-506, 507  
 malloc() function, 49, 130, 131, 132  
 Manipulators, I/O  
   creating custom, 309-312  
   and files, 338, 340  
   table of, 288  
   using, 287-291  
 map class, 476, 502-509  
   member functions, table of, 503-505  
 math.h header file, 10  
 Member functions, 21, 22, 108  
   const, 455-458  
   defining, general form for, 22  
   and this pointer, 126, 128, 129  
 Member variables, public, 25-26  
 Members, class, 21  
   fully specifying, 112-113  
   static, 449-455  
 merge() function, 491, 493, 498-499  
 modf() function, 145  
 multimap, 502  
 mutable keyword, 456, 457-458

**N**

name() function, 409  
 Namespace, 9, 11-12, 437-445  
   unnamed, 439-440, 445  
 namespace statement, 8, 12, 437  
 <new> header, 401  
 new operator, 130-133  
   and allocation failures, 130-131, 401-402

and dynamically allocated arrays, 134, 136-137  
 handling exceptions thrown by, 401-404  
 to initialize dynamically allocated objects, 134, 135-136

nboolalpha I/O manipulator, 290  
 nshowbase I/O manipulator, 288  
 nshowpoint I/O manipulator, 288  
 nshowpos I/O manipulator, 288  
 noskipws I/O manipulator, 288  
 nothrow, 402, 403-404  
 nunitbuf I/O manipulator, 288  
 nouppercase I/O manipulator, 288  
 npos constant, 522, 524

**O**

## Object(s)

arrays of, 119-123  
 assigning, 89-94  
 constructing, "on the fly," 58  
 creating, 22-23  
 data in, 23, 28  
 definition of, 4-5  
 factory, 414  
 to functions, passing, 96-101  
 from functions, returning, 102-106  
 and inheritance, 6  
 initializing. *See* Initialization of objects  
 obtaining address of, 66  
 by reference, passing, 146-148  
 referencing public members of, 23  
 Object-oriented programming (OOP), 2, 3-7  
 Object pointers, 66-67, 124  
 pointer arithmetic and, 125-126  
 oct  
 format flag, 275, 276, 279  
 I/O manipulator, 288  
 off\_type, 331  
 ofstream class, 275, 313, 314  
 open( ) function, 313-315, 319-320  
 Operator functions  
 definition of, 197

friend, using, 213-218  
 member, general form of, 197  
 Operator overloading, 6  
 and the array subscripting operator ( [ ] ), 222-226  
 and the assignment operator ( = ), 213, 218-221  
 basics, 197-199  
 binary, 199-211  
 and built-in types, 204-205, 214-216  
 increment ( + + ) and decrement ( - - ), postfix and prefix, 210-211, 217-218  
 I/O, 292-302  
 relational and logical, 207-208  
 unary, 209-212  
 Operators and null-terminated strings, Standard C++, 519-520  
 Operators, I/O, 13-14  
 overloading, 292-302  
 ostream class, 275, 292, 313  
 ostrstream class, 466  
 Output operator ( < < ), 13-14, 292, 338  
 overload keyword, 39, 177  
 Overloading  
 function. *See* Function overloading  
 operator. *See* Operator overloading  
 Overriding virtual functions, 351-357

**P**

pair class, 478, 505-506  
 Parameter list, declaring empty, 28-29  
 pcount( ) function, 466-467  
 peek( ) function, 328, 329-331  
 Pointer declarations and overloaded functions, 189-191  
 Pointer parameters vs. reference parameters, 140-142  
 Pointers  
 to derived classes, 347-349  
 to objects. *See* Object pointers  
 Polymorphism, 4, 5-6, 7, 409  
 applying, 362-368

and function overloading, 34-36  
 and virtual functions, 346, 349-350, 362-368  
 pos\_type, 332  
 precision() function, 283-285  
 Predicate functions, 477-478  
 printf() and C++, 13  
 private access specifier, 62, 68, 69, 234-235  
 protected access specifier, 62, 234, 240  
 Protected class members, using, 240-243  
 Prototypes, 29, 30-31  
 public access specifier, 21, 62, 234-235  
 push\_back() function, 482, 484, 485, 491, 493, 495  
 push\_front() function, 491, 493  
 put() function, 321, 322-323  
 Put pointer, 332-333  
 putback() function, 328, 329-331

**Q**

queue class, 476

**R**

rand() function, 354, 414  
 Random access I/O, 331-334  
 Random events, responding to at run time, 354-355, 366-368  
 rdstate() function, 335, 336-337  
 read() function, 321, 324-325  
 Reference parameters, 140-145, 220-221  
   and ambiguity, 187-188  
   and friend operator functions, 216-217  
   and member operator functions, 205-206  
 References, 140-155, 220-221  
   independent, 154-155  
   parameter. *See* Reference parameters  
   restrictions on, 154  
   returning, 149-153  
   used to pass objects, 146-148  
 register variables, C versus C++, 532  
 reinterpret\_cast, 429-430  
 Relational operators, overloading, 207-208

remove\_copy() algorithm, 511, 514-515  
 replace() function, 523, 527-528  
 resetiosflags() I/O manipulator, 288, 289  
 return statement, 29  
 reverse() algorithm, 512, 515-516  
 rfind() function, 524  
 right  
   format flag, 275, 276  
   I/O manipulator, 288  
 Run-time type identification (RTTI), 409-419

**S**

scanf() and C++, 13  
 scientific  
   format flag, 275, 276  
   I/O manipulator, 288  
 Scope resolution operator (::), 22, 112-113, 438  
 seek\_dir enumeration, 331  
 seekg() function, 331-332, 333-334  
 seekp() function, 331-333  
 setbase() I/O manipulator, 288  
 setf() function, 277, 289  
 setfill() I/O manipulator, 288  
 setiosflags() I/O manipulator, 288, 289  
 setprecision() I/O manipulator, 288  
 setw() I/O manipulator, 287, 288  
 showbase  
   format flag, 275, 276  
   I/O manipulator, 288  
 showpoint  
   format flag, 275, 276  
   I/O manipulator, 288  
 showpos  
   format flag, 275, 276, 277, 279  
   I/O manipulator, 288  
 size() function, 482, 485, 525  
 skipws  
   format flag, 275, 276  
   I/O manipulator, 288  
 sort() function, 493, 497  
 splice() function, 491, 493-494  
 static class members, 449-455

static\_cast, 429, 431  
 std namespace, 11, 437, 442-444  
 stderr stream, 273  
 stdin stream, 273  
 stdio.h header file, 10  
 stdout stream, 13, 273  
 STL (Standard Template Library), 380, 474, 476-478  
 strcat() function, 522  
 strcpy() function, 449, 520, 522  
 streambuf class, 274  
 Streams (I/O), 273-274, 313  
 streamsize type, 284, 321  
 string class, 474, 519-525  
   operators defined for, 521-522  
 String handling in C++, 519-520, 525-527  
 <string> header, 521  
 string.h header file, 10  
 stringstream class, 466, 467  
 <stringstream> header, 466  
 struct keyword, 68-69, 71  
 Structured programming, 3  
 Structures in C++, 68-74

## I

tell() function, 332  
 tellp() function, 332  
 Template functions. *See* Generic functions  
 template statement, 373, 375-376, 380  
 Templates, 372  
 terminate() function, 388, 395  
 this pointer, 126-129  
   and friend operator functions, 213  
   and inserters, 292-293  
   and member operator functions, 199  
 throw, 386-388, 395  
   statement, general form for, 388  
   clause, general form for, 394-395  
 time\_t type, 165  
 transform() algorithm, 512, 516-518  
 True and false in C and C++, 30  
 true value, 30, 207

try statement, 386-388, 389-392, 398  
 Type conversions and ambiguity, automatic, 185-187  
 typeid operator, 409, 410-419, 422, 426  
 type\_info, 409, 413  
 <typeinfo> header, 409  
 typename keyword, 374, 376

## U

Unary operators, overloading, 209-212  
 unexpected() function, 395, 398  
 Unions, 69-70, 72-75  
 unitbuf  
   format flag, 275, 276  
   I/O manipulator, 288  
 unsetf() function, 277, 289  
 uppercase  
   format flag, 275, 276, 279  
   I/O manipulator, 288  
 using statement, 438-439  
 <utility> header, 478

## V

Variables  
   global, declaring, 532  
   global vs. static, 450  
   local, declaring, 29-30, 31  
   objects as, 5  
   public member, 25-26  
 vector class, 476, 479, 480-490  
   member functions, table of, 482-483  
   using iterator to access, 486-487  
 Virtual base class, 259-261  
 Virtual functions, 349-361  
   hierarchical nature of, 352-354, 361  
   and late binding, 363  
   overriding, 351-357  
   and polymorphism, 346, 349-350, 362-368  
   pure, 358-360  
   responding to run-time random events with, 354-355, 366-368



virtual keyword, 260, 349  
void keyword, 28-29, 30

**W**

wcerr stream, 273-274  
wchar\_t keyword, 532  
wcin stream, 273-274  
wclog stream, 273-274

wcout stream, 273-274  
width() function, 283-284  
write() function, 321, 323-324  
ws I/O manipulator, 288  
wstring class, 519

**X**

xalloc exception, 401