Index

α, 379
π, 27, 93, 403
-unboxing, 74

abstract base class, 149, 151, 160, 408, 505
`AbstractGenerator` example, 152
`AbstractCard` example, 156
abstraction, 5, 6, 151
  data, 5
  procedural, 5
access, random, 179
accessors, 13
adjacency list, 416
adjacency matrix, 410
adjacent, 404
algorithm, greedy, 429
allocation, dynamic, 179
Amend, Bill, 277
analysis
  amortized, 229, 329, 333
  asymptotic, 81–92
  average-case, 90
  best-case, 90
  big-O, 81, 82
  worst-case, 90
`Analysis` example, 85
ancestor, 277
  proper, 279
and
  logical, 262
  short-circuiting, 262
and-gate, 333
API, 45
Application Program Interface, 45
application, scalable, 43
array, 43
  associative, 369
  two-dimensional, 411
assertion, 33–37
association, 14, 369
associative array, 369
associative memory, 66
associative structure, 345–348
asymptotic, 92
asymptotic analysis, 81–92
`atinlay` example, 15
autoboxed, 74
autoboxing, 74, 493

balloon, helium-filled, 189
bank account, 11
`BankAccount` example, 11
base case, recursive, 95
base type, 409
baseclass, 504
big-Ω, 112
big-O notation, 81–92
binary search, 259–264
binary search tree, 343
binary tree, 279
  complete, 280
  full, 279
bound, upper, 44
boundary cases, 212
breadth-first, 251
bubble sort, 120
`BubbleSort` example, 119

C++, xi, 253
cache, 99
`Calc` example, 282
call
  frame, 222
  stack, 222
`Card` example, 155
Carroll, Lewis, 219
cast, 143, 256, 492
child, 279
  left, 279
  right, 279
`Circuit` example, 334
class, 5, 8
  abstract, 149, 191, 408–410, 505
  concrete, 408, 505
  dynamic loading of, 507
  encapsulation, 507
  extension, 408
  hierarchy, 513
  importing, 11
  instance of, 5, 8
  program is a, 489

class cast exception, 143
clone, of object, 211
clustering, 377
  primary, 377, 378
  secondary, 377
code reuse, 235
CoinPuzzle example, 231
collection classes, 511
collections, 511–512
comment, 33–37
  political, 2
comparable, 253
  association, 256
  ratio, 254
comparator, 140–142
compare, 253
comparing values, 197
compile-time, 69
CompInsSort example, 142
complexity, 1, 81
  of method, 235
  space, 82–92
  time, 82–92
component, 404
  connected, 404
  strongly connected, 404
concrete class, 505
connected component, 404
ConstantGenerator example, 153
constructive toys, 1
constructor, 10
  copy, 59, 373
container types, 72
contract, 6
  interface viewed as, 6, 25, 181
control
  flow of, 222
  structures, 8, 161
correctness, proof of, 102
craftsmen, 81
cycle, 404

DAG, 404
Dahl, Roald, 315
data
  comparing, 253
  declaring protected, 25, 515
data abstraction, 6–7
data structure, 1, 8
data types, generic, 72
data types, parameterized, 72
dead code, 91
degree, 279
deprecated, xiii
descendant, 279
  proper, 279
design
  method, 149–160
  properties of, 108–110
deterministic, 134
dictionary, 369
Dijkstra example, 432
disjoint, 277
documentation, 513–516
Dodgeson, Charles Lutwidge, 219
doubly linked list, 201
Doyle, Arthur Conan, Sir, 441
Drake, Milton, 489
dummy nodes, 215
dynamic programming, 99

edge
  canonical, 418
  of tree, 277
  weighted, 428
element
  current, 164
encapsulation, 6–8
  class, 507
  package, 507
encoding, 241, 303
  unicode, 492
energy, 195, 201, 258
enumeration, 161
  live, 162
equals, default, 13
example
  AbstractGenerator, 152
  AbstractCard, 156
  Analysis, 85
  atinlay, 15
  BankAccount, 11
  BubbleSort, 119
  Calc, 282
  Card, 155
  Circuit, 334
  CoinPuzzle, 231
  CompInsSort, 142
  ConstantGenerator, 153
  Dijkstra, 432

    `Example`, 2
    `Fibo`, 106
    `Floyd`, 428
    `FullPostage`, 99
    `Generator`, 152
    `HelloWorld`, 162, 164
    `HexBoard`, 313
    `HexMove`, 313
    `HTML`, 251
    `Huffman`, 303, 304, 317
    `Huffman2`, 317
    `Index`, 396
    `InfiniteQuestions`, 288
    `InsertionSort`, 125
    `LinkedList`, 216
    `LongWords`, 70
    `LongWords2`, 71
    `LongWords3`, 78
    `LSystem`, 56
    `MCST`, 430
    `MergeSort`, 127
    `nim`, xii
    `ParkingLot`, 183
    `ParkingLot2`, 271
    `Pedigree`, 280
    `PFGenerator`, 168
    `PhoneBook`, 138, 143
    `PinochleCard`, 159
    `Player`, 314
    `PokerCard`, 158
    `PrimeGenerator`, 154
    `QuickSort`, 131, 222
    `RadixSort`, 135
    `Ratio`, 8, 254
    `RBSymTab`, 362
    `Reachability`, 422
    `Reader`, 248
    `Rect`, 20
    `Recursion`, 94, 102
    `RecursiveIterators`, 298
    `RecursivePostage`, 98
    `SelectionSort`, 122
    `Sort`, 265
    `sqrt`, 34
    `StringReader`, 43
    `StringVector`, 71
    `SymbolTable`, 249
    `SymMap`, 369
    `SymTab`, 346
    `Token`, 248
    `TopoSort`, 426
    `Unique`, 182
    `UniqueFilter`, 170
    `Warshall`, 427
    `WordFreq`, 48
    `WordList`, 19, 23, 47
`Example` example, 2
example, icon for, xii
extensible, 70
extension of class, 408

`Fibo` example, 106
Fibonacci numbers, 106
fields, 8
FIFO, *see also* queue
filter, 170
finger method, 192
`Floyd` example, 428
foreach, 500
forest, 277
format, 496
Fox, Jason, 277
free list, 183–186, 410
friction, 110
    definition, 108, 110
    in heaps, 320
Frost, Robert, 507
`FullPostage` example, 99
function, *see also* method
functions, static, 322

garbage collection, 199
garbage collector, 189, 494
gate delay, 334
Geisel, Theodor Seuss, 5, 369
generator, 152–155
    as iterator, 167–170
`Generator` example, 152
generic, xv
generic class, 69, 72
Gillespie, Haven, 179
grammars, 57
graph, 403–434
    acyclic, 404
    adjacency list, 416
    arc, *see also* graph, edge
    complete, 404
    dense, 416
    directed, 403
    directed acyclic, 404

edge, 403
node, *see also* graph, vertex
sparse, 416
sub-, 403
transitive closure, 427
undirected, 403
vertex, 403
Guthrie, Arlo, 69

Hangman, 18
hash code, 385–392
hash function
perfect, 377
rehashing, 377
hash table, 374–392
bucket, 375, 376
collision, 377
double hashing, 378
external chaining, 383–385, 392
linear probing, 378
load factor, 379
open addressing, 375–383, 392
ordered linear probing, 400
primary clustering, 378
probe count, 392
rehashing, 377
reserved value, 377
hashing, 375
heap, 308, 319
complete, 320
skew, 329
height, 279
black, 361
of node, 279
of tree, 279
`HelloWorld` example, 162, 164
`HexBoard` example, 313
`HexMove` example, 313
hierarchy, 513–515
`HTML` example, 251
Huffman encoding, 303
`Huffman` example, 303, 304, 317
`Huffman2` example, 317

icons, marginal, xii
implementation, 6, 19
point-of-view, 507
incident, edge to vertex, 403
`Index` example, 396
induction, mathematical, 81, 101–106
`InfiniteQuestions` example, 288
infix, 290
information theoretic limit, 91
inheritance, 408
multiple, 506
insertion sort, 125–127
`InsertionSort` example, 125
instantiation, 442
interface, 5, 6, 22–24, 149, 504
abiding by, 25, 515
design of, 149
implementation, 190
implementing, 23
list, 180
point-of-view, 507
program, 19
Internet, *see also* Web
intractable, 85
inverter, 333
iterator, 161–173
filtering, 170–172
is read-only, 211
potato, 161

Java, xi, 1, 5, 489–506
jit compiler, 117
Johnson, Crockett, 149
just-in-time compiler, 117

key
in association, 369

L-systems, 56
language, purpose of, 69
Lao Tzu, 81
larger, 123
length of path, 279
level, 279
LIFO, *see also* stack
linear structures, 219–246
`LinkedList` example, 216
list, 179–211
abstract, 186
add, 179
adjacency, 416–422
circular, 206–209
concept, 179
doubly linked, 201–206
head, 179
interface, 180

- is a graph, 403
- iterator, 209–211
- remove, 179
- singly linked, 188–201
- tail, 179

load factor, 383
logic gates, 333
`LongWords` example, 70
`LongWords2` example, 71
`LongWords3` example, 78
`LSystem` example, 56

map, 369–398
- ordered, 392–398

Mars, life on, 191
matrix, 60–63
- adjacency, 410–416
- symmetric, 66, 410
- triangular, 66

maze
- finish cell, 242
- start cell, 242

mazes
- solving, 242–244

McCloskey, Robert, 119
`MCST` example, 430
mechanics of data structures, 6
median, 259
memory leak, 194
mention, 403
merge, 127
mergesort, 127–131
`MergeSort` example, 127
message, 26, *see also* method
method, 8, 490
- calling, 5
- calls, 8
- destructive, 213, 330
- helper, 318
- nondestructive, 194
- postcondition, xii
- precondition, xii
- `this`, 17

methods, static, 322
minimum spanning tree, 429
Modula-2, xi
mutator, 13

nanosecond, 93, 115
Napster, 15
`new` operator, 12, 188
`nim` example, xii
node
- degree, 279
- depth, 279
- interior, 279
- level, 279
- parent, in tree, 277
- sibling, 279
- of tree, 277

nondeterministic, 134
`null`, 189

Oakland, Ben, 489
object, 5, 8, 489
- general, 345
- model of, 7–8

object orientation, 5–25
- in languages, 5
- terminology of, 8–11

object-oriented programming, 5–25
objects, comparable, 253–258
octtrees, 308
optimization, 91
ordered
- list, 267
- map, 392–398
- vector, 259

ordered structure, 144
ordered structures, 253–272
overriding methods, 357
overriding operators, 253

package, 11, 489
- default, 509
- encapsulation, 507
- user, 509

`ParkingLot` example, 183
`ParkingLot2` example, 271
partially ordered, 140
Pascal, xi
path, 279, 404
- end points, 404
- simple, 404

`Pedigree` example, 280
performance
- exponential, 85
- linear, 83
- polynomial, 83
- superlinear, 83

- `PFGenerator` example, 168
- philosophy, 1–3
- `PhoneBook` example, 138, 143
- Pig Latin, 14
- `PinochleCard` example, 159
- pivot, 131, 133
- `Player` example, 314
- pointer, *see also* reference
- `PokerCard` example, 158
- postcondition, xii, 34
- postfix, 290
- precedence, operator, 282
- precondition, xii, 34
- `PrimeGenerator` example, 154
- principle, xii, 3
  - assertions, 35, 515
  - code reuse, 17
  - code review, 269
  - consistent interfaces, 51
  - consistent state, 191
  - drawing pictures, 189
  - equal hash codes, 386
  - fighting imperfection, 509
  - hiding opposing references, 286
  - icon for, xii
  - interface design, 25
  - live enumerations, 162
  - most general method, 300
  - natural interface, 319
  - overriding signatures, 256
  - principled programmer, 3
  - progress in recursion, 96
  - protected data, 25
  - providing hash function, 385
  - public protection, 508
  - questioning asymmetry, 201
  - read-only iterator values, 211
  - reducing friction, 320
  - static functions, 322
  - symmetry, 194
  - testing boundaries, 197
  - understanding complexity, 235
- principles, 1–3, 515–516
- principles, abstract, xi
- priority queue, 315
- problems, importance of, 2
- procedure, 249, *see also* method
- productions, 56
- program counter, 222
- program, state, 34
- programming, craft of, xi
- programs, writing large, 1
- progress in recursion, 95
- proof by induction, 94
- `protected`, use of keyword, 507–509
- protection of fields, 8, 10

- queue, 220, 229–242
  - dequeue, 229
  - enqueue, 229
  - head, 229
  - priority, 315
  - tail, 229
- quicksort, 131–134
- `QuickSort` example, 131, 222

- `RadixSort` example, 135
- random access property, 45
- `Ratio` example, 8, 254
- `RBSymTab` example, 362
- `Reachability` example, 422
- reachability, computing, 427
- `Reader` example, 248
- record, traditional, 8
- `Rect` example, 20
- recursion, 81, 94–101
- `Recursion` example, 94, 102
- recursive, 94
- recursive data structure, 310
- `RecursiveIterators` example, 298
- `RecursivePostage` example, 98
- reference, 19, 188, 189, 493
  - null, 189
- rewriting, 56
- Rhoda, 203
- Rhonda, 203
- rising edge, 337
- rotations, 354
- runtime, 44, 69

- scope, of identifier, 491
- search
  - binary, 260
  - depth-first, 422
  - linear, 268
- selection sort, 122–123
- `SelectionSort` example, 122
- self-reference, 81, 94–106
- semantics, 69, 369
  - of data structure, 6

  - implementation independent, 181
  - implementation specific, 182
- Seuss, Dr., 14
- shallow copy, 311
- Shelley, Percy Bysshe, 513
- signature, 19, 149
- Slobodkina, Esphyr, 343
- solution
  - self-referential, 94
- sort
  - bucket, 134
  - radix, 134–138
- `Sort` example, 265
- sorting, 119–144, 264, 345, 424
  - objects, 138–140
  - vectors, 143–144
- Soundex, 401
- splay operation, 354
- splay tree, 354
- `sqrt` example, 34
- stability
  - of sort, 146
- stack, 220–229
  - trace of, 36
- start string, 56
- state, 222
  - program, 34
  - structure, 150
- string
  - counted, 7
  - end mark, 7
  - terminated, 7
- `StringReader` example, 43
- `StringVector` example, 71
- structure
  - control, 8, 161
  - data, 8
  - deterministic, 354
  - dynamic, 179
  - linear, 219, 277
  - nonlinear, 277
  - symmetric, 201
- `structure` package, 11
  - `AbstractIterator`, 163
  - `AbstractLinear`, 220
  - `AbstractList`, 186
  - `AbstractQueue`, 230
  - `AbstractStructure`, 391
  - `Assert`, 35
  - `Association`, 15, 72
- `Association`, 15–17
- `BinarySearchTree`, 348
- `BinarySearchTree`, 344
- `BinaryTree`, 284, 291, 300, 392
- `BinaryTreeNode`, 355
- `BTInorderIterator`, 294
- `BTLevelorderIterator`, 296
- `BTPostorderIterator`, 295
- `BTPreorderIterator`, 291
- `CaselessComparator`, 141
- `ChainedHashTable`, 384
- `CircularList`, 207
- `Comparable`, 254
- `ComparableAssociation`, 256
- `Comparator`, 141
- documentation of, 2
- `DoublyLinkedList`, 203
- `DoublyLinkedNode`, 203
- downloading, xii, 1
- `Edge`, 407
- `Enumeration`, 161
- `Graph`, 404
- `GraphListDirected`, 420
- `GraphListUndirected`, 419
- `GraphListVertex`, 416
- `GraphMatrix`, 408, 411, 412, 414
- `GraphMatrixDirected`, 409, 412, 413
- `GraphMatrixUndirected`, 413, 414, 416
- `Hashtable`, 375, 383
- icon for, xii
- `Iterator`, 163
- `Linear`, 219
- `List`, xii, 180
- `Map`, 370
- `MapList`, 372
- `Matrix`, 60
- `MazeRunner`, 242
- `NaturalComparator`, 266
- `Node`, 189
- `OrderedMap`, 393
- `OrderedStructure`, 259
- `OrderedVector`, 259
- `OrderedList`, 267
- `PriorityVector`, 318
- `PriorityQueue`, 315
- `Queue`, 229
- `QueueArray`, 239
- `QueueList`, 234

    QueueVector, 237
    ReadStream, 494
    ReverseComparator, 266
    Set, 23
    SetVector, 57
    SinglyLinkedList, 191
    SinglyLinkedListIterator, 210
    SkewHeap, 329
    SplayTree, 357
    SplayTreeIterator, 358
    Stack, 221
    StackList, 227
    StackVector, 225
    Structure, 2, 22
    Table, 393
    ValueIterator, 382
    Vector, 45, 50, 74, 96, 151, 165
    VectorIterator, 165
    VectorHeap, 321
    Vertex, 407
structures
    associative, 345
subclass, 409, 504
subroutine, *see also* method
subtype, 409, 503–504
superclass, 409
supertype, 504
SymbolTable, 346
SymbolTable example, 249
SymMap example, 369
symmetry, 108–110, 201
    in interface, 108
    is predictability, 108
SymTab example, 346
syntax, 69
Syracuse sequence, 113

table, 392–398
tail recursion, 98
the entire argument array, 70
The Phone Company, 1
this, 17
time stamp, 424
time–space trade-off, 193
token, 247
Token example, 248
topological sort, 424
TopoSort example, 426
totally ordered, 140
tree, 277–309
    arity, 279
    AVL, 311
    binary, 279
    binary search, 343–364
    complete, 278
    degenerate, 279
    degree, 279
    expression, 282
    full, 278, 279
    height balanced, 311
    is a graph, 403
    leaf, 279
    minimum spanning, 429
    oriented, 279
    pruning, 313
    red-black, 361
    root, 277
    rotation, 354
    splay, 354–360
    sub-, 277, 279
    traversing expression, 282
    trivial, 277
type
    conversion, 492
    primitive, 491
    reference, 491
type bound, 79
type parameter
    actual, 76
    formal, 75
type parameters, 73

Unique example, 182
UniqueFilter example, 170
user, 24

value
    in association, 369
value, comparison of, 197
vector, 43–64
    capacity, 46, 50, 65
    cost of heapifying, 339
    end, 47
    extending, 53
    extent of, 50
    iterator, 165–167
    size, 65
    size of, 50
    tail, 65
    two-dimensional, 60

vertex
- degree, 403
- destination, 403
- in-degree, 403
- out-degree, 403
- sink, 403
- source, 403

`Warshall` example, 427
Warshall's algorithm, 427
Web
- programming, xi
- resources available on, 2

White, Elwyn Brooks, 253
`WordFreq` example, 48
`WordList` example, 19, 23, 47