

# Index

**Note to the reader:** Throughout this index **boldfaced** page numbers indicate primary discussions of a topic. *Italicized* page numbers indicate illustrations.

---

## A

- abstract classes, **138–143**
  - declaring, 138–143, 166
  - defined, 138
- abstract keyword, 138, 140
- abstract methods, 140–143
  - child classes and, 140, 141, 166, 410
  - overriding, 410
  - private methods and, 398
- abstract modifier, **397–399**, 410
- abstraction, 139. *See also* object-oriented programming
- access modifiers (access specifiers), **393–397**, 410. *See also* default access;  
private; protected; public inheritance and, 393–397, 410 local inner classes and, 158 for methods, 116 nested classes and, 163 for top-level classes, 4
- accessor methods (getter methods), 119, 165, 382, 387, 423
- ActionListener, 17, 161
- additive operators, 44, 46–47
- Address class, 385, 386, 387, 392, 393
- Address2 class, 386, 387
- addScore method, 271, 272, 273
- AND operator (&), 44, **53–54**
  - logical (&&), 44, 53, 54
- anonymous inner classes, **159–162**
  - interfaces and, 161
  - local inner classes *v.*, 161
- API (Application Programming Interface, Java), 4
  - Collections, 89, 203, 270, 302, 425, 426
    - documentation, 5
  - append method, 275, 278, 279, 280, 281, 288, 325
  - Apple class, 82, 101, 102
  - Application Programming Interface. *See* API
  - arguments. *See also* command-line arguments; variable-length arguments defined, 36 passing, *by value*, 36
  - arithmetic operators, **46–50**
    - additive, 44, 46–47
    - decrement (--), 44, 49–50
    - increment (++), 44, 49–50
    - multiplicative, 48–49
  - ArithmeticException, 54, 225, 226, 227, 228, 232, 234, 237, 267
  - arrays, **88–96**, 167, **470–475**. *See also specific arrays*
    - command-line arguments and, 20–21
    - conversions
      - arrays into lists, 472
      - lists into arrays, 466–467
    - declaring, 88–96
    - defined, 88, 165
    - enhanced for loops and. *See* enhanced for loops
    - length attribute and, 243
    - maps *v.*, 445
    - multidimensional, 93–95
    - searching, 473–475, 477
    - sorting, 470–473, 477
    - square brackets ([ ]) and, 21, 89, 93, 294
    - using, 91–93

- array initializers, 95–96
  - array objects, 90–91
  - array references, 89
    - array objects and, 90, 91
    - declaring, 89
  - ArrayDemo program, 92, 93
  - ArrayDeque, 430, 475
  - ArrayIndexOutOfBoundsException, 21, 167, 242, 243
  - ArrayList, 428, 440, 475, 476
  - ArrayList object, 26, 27
  - ASCII characters, 23, 78
  - assert statement, 219
  - AssertionError, 219, 222, 232, 242, 247, 249, 266
  - assertions, 188, 218–224, 249, 250
    - class invariants and, 221, 223
    - code outcome and, 222
    - control flow invariants and, 221, 222
    - enabling, 220
    - exceptions *v.*, 224
    - internal invariants and, 221
    - purpose of, 221
  - assignment operators, 44, 44–46
    - compound, 44–46
    - simple, 44, 45
  - assignment statements, boolean
    - expressions and, 191
  - atomic manner, 357
  - atomic tasks, 362
  - autoboxing, 131, 271, 272–274, 325
  - average method, 128, 457
    - on inner classes, 155
    - of instance methods, 124
  - binarySearch methods
    - for arrays, 473–475, 477
    - for lists, 461, 467–469
  - bitwise operators, 44, 53–55
    - AND (&), 44, 53–54
    - exclusive OR (^), 44, 53, 54
    - inclusive OR (|), 44, 53, 54
  - blank finals, 399
  - blocked threads, 349, 353, 368, 369
  - blueprints, classes as, 2, 85
  - Book class, 136, 137
  - boolean (primitive type), 22, 51, 190
  - Boolean (wrapper class), 270
  - boolean add(E e), public, 427
  - boolean contains(Object e), boolean, 427
  - boolean expressions
    - assignment statements and, 191
    - if statements and, 190
    - numeric types and, 51
  - Boolean objects, 463
  - Boolean operator, 406. *See also* instance of operator
  - boolean remove(Object e), remove, 427
  - bounded generic types, 455–457
  - bounded type parameters, 455
  - Box class, 163, 453, 454
  - braces/brackets. *See* curly braces; parentheses; square brackets
  - break statements, 211–215
    - examples, 211, 212, 213
    - labeled, 213
    - syntax, 212
  - Breakable, 451, 452
  - breathe method, 137, 138
  - Buffalo class, 142
  - BufferedInputStream, 282, 284, 287, 324
  - BufferedOutputStream, 282, 284, 287, 293
  - BufferedReader class, 282, 283, 284, 285, 286, 287, 324
- 
- B**
- %b, 297, 325
  - backslash (\), 319. *See also* character classes
  - BankAccount class, 358, 359, 360, 361
  - basic for loops. *See* for loops
  - behind-the-scene details

BufferedWriter class, 283, 284, 285, 287, 289, 324  
 Button objects, in ColorChanger class, 114  
 ButtonListener class, 404, 405  
 Byte, 192, 194, 270, 271, 463  
 byte, 22  
 byte streams. *See* FileInputStream class;  
 FileOutputStream class  
 ByteArrayInputStream, 282, 284  
 bytecode  
   defined, 3  
   /source code, separating, 19

---

## C

%c, 297, 325  
 call by value, 36–43, 62  
 call stack, 42  
 Camera class, 103–104  
 canExecute, 287  
 canRead, 287  
 canWrite, 287  
 Car class, 112, 113  
 case statements, 149, 184, 192, 193, 194, 195. *See also* switch statements  
 case values, final, 196–197  
 casting. *See* converting  
 Cat class, 390, 402, 403, 404, 405, 406  
 catch clauses (exception handlers), 227, 229–231. *See also* try statements  
   multiple, 229–231  
   order of, 231  
 catching errors/runtime exceptions, 233  
 CellPhone class, 395  
 chaining streams  
   BufferedReader/FileReader and, 285, 286, 286  
   FileInputStream/FileOutputStream and, 289, 290  
   FileWriter/BufferedWriter/PrintWriter and, 287, 288, 289, 289  
 char, 22  
 Character (wrapper class), 192, 194, 270  
 character classes, 319–320, 320. *See also* metacharacters  
   \d, 269, 316, 319, 320  
   \s, 269, 316, 319, 320  
   \w, 269, 316, 319, 320  
 character streams. *See* FileReader class;  
 FileWriter class  
 CharacterSorter program, 462–463  
 CharArrayReader, 282  
 CharArrayWriter, 283, 284  
 checked exceptions, 232, 232, 237. *See also* ClassNotFoundException;  
 InterruptedException; IOException  
 CheckedDemo class, 234, 235, 236, 237  
 child classes  
   abstract methods and, 140, 141, 166, 410  
   final methods and, 137, 138, 410  
   is-as relationship and, 390, 391, 410  
   method hiding and, 135, 136, 166  
   method overloading and, 128, 129, 166  
   method overriding and, 165  
   polymorphism and, 402, 403, 410, 411  
   protected modifier and, 393, 410  
 ChildListener class, 404  
 Circle class, 159  
 .class file, 3, 11, 97, 115, 143, 165  
 class invariants, 221, 223  
 class loader, 115  
 class methods. *See* static methods  
 class variables. *See* static fields  
 ClassCastException, 52, 232, 242, 244, 406, 407, 437, 461, 464. *See also* instanceof operator  
 classes, 97–100. *See also* inheritance;  
   inner classes; methods; nested classes;  
   objects; *specific classes*  
   abstract, 138–143, 166  
   as blueprints, 2, 85  
   child. *See* child classes

- classes (*continued*)
  - cohesion and, 388–390, 409
  - composition and, 390, 392, 409, 410
  - cookie analogy and, 2
  - coupling and, 385–388, 409
  - declaring, 97–100
  - defined, 2, 97
  - elements in, 97, 165
  - generic, 450–451
  - house analogy and, 2, 85
  - inner, 152
  - interfaces *v.*, 143
  - multiple, in single file, 4, 62
  - in packages, 5–6, 61. *See also* packages
  - parent. *See* parent class
  - top-level, access modifiers for, 4
  - wrapper, 270–272
  - writing, 2–4
- ClassNotFoundException, 73, 232, 234, 235, 236, 237, 304
- classpath, 11
- CLASSPATH environment variable, 11, 61
- classpath flag, 17
- clickCount method, 87, 88
- code outcome, assertions and, 222
- code output, 61
- code reuse. *See* reuse
- cohesion, 388–390, 409
- Collection interface, 426, 427, 475.
  - See also* Collections class; collections interfaces
  - methods in, 427
  - subinterfaces in, 427
- collections, 426–435. *See also* generics
  - classes, list of, 476
  - defined, 426
  - enhanced for loops and, 203–204
  - heterogeneous, 409
  - nongeneric, limitations of, 436–437
  - overview of, 426
  - review questions, 478–489
  - Collections API, 89, 203, 270, 302, 425, 426
  - Collections class, 461, 467, 469, 476, 477, 487. *See also* Collection interface; searching lists; sorting lists
  - Collections Framework (Java), 426, 475
    - equals method and, 434. *See also* equals method
    - generics and, 427, 437, 476. *See also* generics
  - collections interfaces, 426–432, 476
    - lists, 426, 427–428, 438–441, 476
    - maps, 426, 430–432, 476
    - queues, 426, 429–430, 476
    - sets, 426, 428–429, 441–445, 476
  - ColorChanger class, 97–100
    - Button objects in, 114
    - elements in, 99–100
    - GUI programming and, 17, 100
    - MyButtonListener class, 157
    - running Java from command line and, 14, 15, 16, 17
  - command line
    - arguments, 19–22
      - arrays and, 20–21
      - running Java applications from, 12–22
        - ColorChanger and, 14, 15, 16, 17
        - TestColors and, 15, 16, 17
  - Comparable interface, 432–433
  - comparable sorting, 462–464
  - Comparator object, 461, 467, 468, 470, 473, 475, 477
  - comparators
    - natural ordering *v.*, 477
    - sorting, 465–467
  - compareTo, 432–435
    - consistent with equals and, 435, 476
    - natural ordering and, 433, 435, 461, 462, 476
  - comparison operator (==), 56–58, 62.
    - See also* equals method
    - autoboxing and, 274
    - equals method *v.*, 62, 433–435, 477

- compile method, 315, 325
  - composition, 390, 392, 409, 410. *See also*
    - has-a relationship
  - compound assignment operators, 44–46
  - computeArea method, 398
  - concat method, 275, 325. *See also*
    - String class
  - concatenation
    - + operator and, 46, 47, 276
    - optimizing, 277
  - concrete subclasses, 141
  - concurrency. *See also* threads
    - defined, 342
    - processes and, 342
    - threads and, 342
  - conditional (ternary) operator (? :), 44, 55
  - conditional/logical operators. *See* logical operators
  - consistent with equals, compareTo and, 435, 476
  - Console class, 298–301
    - methods, 299
    - System.in and, 298, 301
    - System.out and, 298, 301
  - Console enum, 195
  - constantexpression, 192, 192
  - constants
    - blank finals as, 399
    - DateFormat class and, 312
    - enumerations and, 147, 166, 196
    - final variables and, 196, 399, 400
    - naming conventions for, 400
  - constructors, 102
    - Camera class and, 103–104
    - default, 104–105
    - defined, 97, 102, 165
    - enum, declaring, 151–152
    - properties, 102
    - Scanner class and, 322
    - super keyword in, 108–111, 166
    - this keyword in, 105–107, 166
  - Consumer class, 364, 365, 366, 367
  - consumer model. *See* producer/consumer model
  - Contact class, 291
  - Contact2 class, 302, 303
  - ContactManager program, 291, 292, 293, 294
  - containers. *See* collections
  - continue statements, 215–218
    - labeled, 217, 218
    - syntax, 216
  - control flow invariants, 221, 222
  - control structures
    - boolean expressions and, 190
    - break statements and, 211–215
    - decision making, 188, 249, 250
    - repetition, 188, 249, 250
  - conversion specifiers. *See* format specifiers
  - convert method, 130
  - converting (casting). *See also*
    - ClassCastException
  - arrays into lists, 472
  - autoboxing and. *See* autoboxing
  - cast operator and, 47
  - lists into arrays, 466–467
  - numbers to strings. *See* concatenation;
    - DecimalFormat class
  - primitive types, 47
  - primitive types into strings, 276
  - references (polymorphic), 47, 405–409
  - strings into dates, 314–315
  - strings into numbers, 310–311
- cookie analogy, classes/objects and, 2
- CopyFile program, 290, 291
- CountToTen program, 345, 346, 352
- coupling, 385–388, 409
- covariant return types, 134–135
- cp flag, 17
- CreateEmployee program, 6, 7
- CreateEmployee2 program, 7
- createNewFile method, 287, 337
- Cupboard class, 450, 451, 456

curly braces ( { } )  
 abstract methods and, 141  
 do statement and, 209  
 if statement and, 189  
 instance initializers and, 112  
 try statement and, 227  
 while statement and, 206

currency  
 formatting, 306, 307, 308, 309  
 parsing, 312

Customer class, 121, 122

---

## D

%d, 297, 325  
 \d (metacharacter), 269, 316, 319, 320  
 -d flag, 6, 10  
 -da flag, 220

data  
 formatting/parsing, 306–315, 324, 325  
 input/output. *See* input/output

data types. *See also* generic types;  
 primitive types; reference types  
 interfaces and, 146  
 promotion, method overloading and,  
 129–130

DataInputStream class, 281, 282, 284,  
 291–294, 324

DataOutputStream class, 281, 282,  
 291–294, 324

DateFormat class, 312–315, 324  
 constants and, 312  
 format methods, 313–314  
 parse method, 314–315  
 static methods, 312–313

dates, strings into, 314–315

dead threads. *See* terminated threads

deadlock, 360–361

DecimalFormat class, 309–310. *See also*  
 NumberFormat class

decision making control structures, 188,  
 249, 250. *See also* if statements;  
 switch statements

declaring  
 abstract classes, 138–143, 166  
 array references, 89  
 arrays, 88–96  
 classes, 97–100  
 enum constructors, 151–152  
 enum methods, 150–152  
 enumerations, 147–152  
 interfaces, 143–147  
 methods, 116–138, 165  
 nested classes, 152–164  
 variables, 78–79, 165

decrement operators (--), 44, 49–50

default access (access modifier), 393, 410.  
*See also* access modifiers  
 classes and, 3  
 constructors and, 102  
 interfaces and, 143  
 member inner class and, 153  
 methods and, 116  
 top-level classes and, 4

default constructors, 104–105  
 super keyword and, 110–111

delete()  
 File class, 287  
 StringBuffer/StringBuilder classes, 280

deleteCharAt(), 280

delimiters, 322, 323, 325

Deque interface, 426, 428, 430, 475

deques, 430, 475. *See also* queues

deserialization, 301. *See also* serialization

DeserializeDemo program, 304,  
 305, 306

design principles, OO. *See* OO design  
 principles

Direction enum, 149, 150, 151

-disableassertions flag, 220

Dish objects, 452, 454

division (/) operator, 48

do statements. *See* do-while loops

Dog class, 406  
 equality of objects in, 59–60  
 Fido object, 60  
 finalize method and, 34, 35

Lassie object, 35, 60  
 naming conventions and, 9  
 Snoopy object, 35, 36  
 DogTest program, 60  
 dollar sign (\$), 78, 155, 183, 297, 309, 312  
 doSomething class, 20  
 doSomething method, 134, 159, 337  
 dot (.), 269, 315, 316, 317  
 Double (wrapper class), 270, 272, 273, 312, 339, 463  
 doubles, 22  
   precision of, 159  
 do-while loops (do statements), 209–211  
   examples, 209, 210  
   rules for, 209  
   variables, scope of, 211  
   while loops *v.*, 208  
 Drawable interface, 144, 145, 146, 147

---

## E

E (element), 452  
 E ceiling(E e), 444  
 E floor(E e), 444  
 ⌊E⌋ generic return type, 427, 437. *See also* generics  
 E higher(E e), 444  
 E lower(E e), 444  
 -ea flag, 220  
 eat method, 132, 133, 134  
 element (E), 452  
 eligible, for garbage collection, 31–32, 62  
 ellipsis. *See* variable-length argument list  
 else block, 189. *See also* if statements  
 Email class, 130, 131  
 Employee class, 106, 107, 119, 120, 386, 391, 392, 393  
 Employee2 class, 387  
 Employee3 class, 388  
 -enableassertions flag, 220  
 encapsulation, 382–385, 409, 423  
 enhanced for loops (for-each loops), 92, 201–205  
   collections and, 203–204  
   for loops *v.*, 204–205  
   nested, 205  
   properties, 202  
   syntax, 201  
   variables, scope of, 202–203  
   when to use, 204–205  
 enums  
   Console, 195  
   defined, 147, 166  
   Direction, 149, 150, 151  
   IceCream, 151, 152  
   Season, 148  
   switch statements and, 149, 194–196  
   using, 149–150  
 enum constructors, 151–152  
 enum keyword, 148, 166, 167  
 enum methods, 150–151  
 enumerations, 147–152, 167  
   constants and, 147, 166, 196  
 equal to. *See* comparison operator  
 equality, of objects, 58–61  
 equality operators, 56–58  
   comparison (==), 56–58, 62  
   autoboxing and, 274  
   equals method *v.*, 62, 477  
   not equal to (!=), 56–58  
 equals method, 58. *See also* comparison operator  
   Collections Framework and, 434  
   comparing strings with, 276  
   comparison operator (==) *v.*, 62, 433–435, 477  
   hashCode method and, 60–61  
 erasure, type, 451  
 errors, 232, 232. *See also* exceptions  
   AssertionError, 219, 222, 232, 242, 247, 249, 266  
   catching, 233  
   ExceptionInInitializerError, 242, 247–248  
   NoClassDefFoundError, 14, 16, 73, 242, 249  
   StackOverflowError, 242, 248  
 escape (\), 319. *See also* character classes

- event handlers, inner classes as, 157, 161–162
  - event listener methods, 120–121
  - exception handlers. *See* catch clauses
  - exception handling clauses, 224
    - catch clauses, 227, 229–231
    - try statements, 227–229
  - ExceptionDemo class, 225
  - ExceptionInInitializerError, 242, 247–248
  - exceptions, 188, 224–249, 250
    - ArithmeticException, 54, 225, 226, 227, 228, 232, 234, 237, 267
    - ArrayIndexOutOfBoundsException, 21, 167, 242, 243
    - AssertionError, 219, 222, 232, 242, 247, 249, 266
    - assertions *v.*, 224
    - categories of, 231–232, 232
    - checked, 232, 232, 237
    - ClassCastException, 52, 232, 242, 244, 406, 407, 437, 461, 464
    - ClassNotFoundException, 73, 232, 234, 235, 236, 237, 304
    - errors, 232, 232
    - ExceptionInInitializerError, 242, 247–248
    - FileNotFoundException, 229, 230, 231, 240, 286, 322, 350, 460, 461
    - Handle or Declare Rule and, 231–237
    - IllegalArgumentException, 224, 242, 245, 351, 384
    - IllegalStateException, 242, 245–246
    - IllegalThreadStateException, 351, 355, 369, 379
    - InterruptedException, 116, 232, 237, 353
    - IOException, 8, 231, 232, 233, 237, 239, 266, 267
    - NoClassDefFoundError, 14, 16, 73, 242, 249
    - NullPointerException, 122, 194, 229, 232, 233, 237, 241, 242, 246, 247, 267, 268, 337
    - NumberFormatException, 242, 246–247, 266, 267, 272
    - runtime, 231, 232, 233, 237
    - RuntimeException, 231, 232, 232, 267
    - StackOverflowError, 242, 248
  - exclusive OR operator (^), 44, 53, 54
  - exists method, 287
  - explicit initialization, 81–83
  - extending
    - interfaces, 147
    - Thread class, 343, 346–348, 368, 369
  - extends keyword, 147, 455, 456
- 
- F**
- %f, 297, 325
  - Favorites program, 204
  - Feline class, 142, 143
  - FictionBook class, 111, 136, 137
  - Fido object, 60
  - fields, 80. *See also* instance variables; static fields
  - FIFO (first-in, first-out), 426, 429, 430. *See also* LIFO; queues
  - File class, 287–289, 324
    - methods of, 287, 325
    - uses of, 287
  - file input/output, 285–301, 324
  - FileDemo program, 288, 289
  - FileInputStream class, 282, 284, 285, 287, 289–291, 324
  - FileNotFoundException, 229, 230, 231, 240, 286, 322, 350, 460, 461
  - FileOutputStream class, 282, 284, 287, 289–291, 324
  - FileReader class, 284, 285–287, 324
  - FileWriter class, 284, 285–287, 324
  - final case values, 196–197
  - final keyword, 137
  - final methods, 137–138, 410
  - final modifier, 399–401, 410

- final variables, 196–197
    - constants and, 196, 399, 400
    - naming convention for, 400
  - finalize method, 33–36
  - finally block, 238–241, 250
    - syntax, 238
    - try statement and, 238–241, 250
  - FinallyDemo program, 238, 239, 240
  - findByName method, 37, 38
  - finishTimes reference, 89, 90
  - first-in, first-out (FIFO), 426, 429, 430.
    - See also* last-in, first-out; queues
  - flag
    - classpath, 17
    - cp, 17
    - d, 6, 10
    - da, 220
    - disableassertions, 220
    - ea, 220
    - enableassertions, 220
  - float, 22
  - Float (wrapper class), 270, 271, 272, 273, 452, 463
  - flow control, 187–249
    - assertions and, 188, 218–224, 249, 250
    - decision making control structures, 188, 249, 250
    - exceptions and, 188, 224–247, 249, 250
    - overview of, 188
    - repetition control structures, 188, 249, 250
      - continue statements and, 215–218
  - Flower class, 146
  - for loops (basic), 197–201. *See also*
    - enhanced for loops
    - enhanced for loops *v.*, 204–205
    - examples, 198, 199, 200
    - length attribute and, 243
    - nested, 199–200
    - optional sections of, 201
    - properties, 197
    - syntax, 198
    - types of, 197
    - variables, scope of, 199
  - for-each loops. *See* enhanced for loops
  - foreign locales, 308, 325
  - formal type parameters, 450, 451, 456
  - format methods
    - DateFormat class, 313–314
    - NumberFormat class, 307–309
    - PrintWriter class, 295–298, 324
  - format specifiers, 296–297, 297, 325
  - Formatter class, 298. *See also*
    - PrintWriter class
  - formatting currency, 306, 307, 308, 309
  - formatting data, 306–315, 324, 325
  - “forms, many.” *See* polymorphism
  - free store, 28. *See also* heap
  - friendly access. *See* default access
  - Fruit class, 101
  - FullyQualifiedDemo program, 9
  - fundamentals (Java), 1–75
- 
- G**
- garbage collection, 28–36, 62
    - eligible for, 31–32, 62
    - finalize method and, 33–36
    - GregorianCalendar objects and, 29–31
    - System.gc method and, 32–33
  - Garfield, 403
  - GCDemo program, 31
  - GCDemo3 program, 35, 36
  - generics, 436–461
    - Collections Framework and, 427, 476
    - ⟨E⟩ generic return type and, 427, 437
    - lists and, 438–441
    - maps and, 445–449
    - naming conventions for, 452–453
    - nongeneric collections and, 436–437
    - polymorphism and, 456, 458, 476
    - review questions, 478–489
    - sets and, 441–445

generics (*continued*)  
     syntax, 450, 454, 455, 476  
     type erasure and, 451  
 generic classes, 450–451  
 generic data type (T), 453  
 generic interfaces, 451–453  
 generic methods, 453–455  
 generic supertypes, 459, 461  
 generic types, 449–461  
     bounded, 455–457  
     naming conventions for, 452–453  
     type erasure and, 451  
     wildcard, 458–460  
 generic wildcards. *See* generic types  
 get methods, 119, 120  
 getBalance method, 359, 378  
 getCounter method, 136, 137  
 getCurrencyInstance(), 307  
 getDateInstance(), 312  
 getDateTimeInstance(), 312, 313  
 getFile method, 43  
 getInstance(), 306  
 getInstance(), 306  
 getMessage(), 229  
 getNumberInstance(), 306, 307  
 getPercentInstance(), 307  
 getter methods (accessor methods), 119,  
     165, 382, 387, 423  
 getTimeInstance(), 312  
 global methods, static methods *v.*, 125  
 global variables, 83, 85. *See also* static  
     fields  
 graphical user interface (GUI)  
     programming, 17, 100  
 greater than (>), 51  
 greater than or equal (>=), 51  
 greedy quantifiers, 315  
 GregorianCalendar  
     objects, garbage collection and, 29–31  
     references, 90, 91, 91  
 GUI (graphical user interface)  
     programming, 17, 100

---

## H

Handle or Declare Rule, 231–237  
 has-a relationship, 392–393, 409  
 hashCode method, 60–61  
 HashMap, 431, 447, 475, 476  
 HashSet, 429, 441–442, 475, 476  
 hasNext methods, 322  
 headMap method, 448  
 headSet method, 444  
 heap, 28  
 Hello class, 455. *See also* SayHello class  
 heterogeneous collections, 409  
 hidden static methods, 137  
 hiding. *See* encapsulation; information  
     hiding; method hiding  
 high cohesion, 388–390, 409. *See also*  
     loose coupling; tight encapsulation  
     loose coupling and, 389  
     tight encapsulation and, 389  
 high-level streams, 283–285, 284, 324  
 house analogy, classes/objects and, 2, 85  
 House class, 83, 85  
 HouseCat class, 143  
 HouseTest program, 84, 85, 86  
 HumanResources class, 389

---

## I

IceCream enum, 151, 152  
 identifiers, 78–79  
     invalid, 79  
     tokens as, 79  
     valid, 79  
 IEEE website, 159  
 if statements (if-else, if-then-else), 188–192  
     boolean expressions, 190  
     curly braces and, 189  
     else block and, 189  
     rules for, 189  
     switch statements *v.*, 192  
     syntax, 189

- if-else statements. *See* if statements
- if-then-else statements. *See* if statements
- IllegalArgumentException, 224, 242, 245, 351, 384
- IllegalStateException, 242, 245–246
- IllegalThreadStateException, 351, 355, 369, 379
- immutable String objects, 25, 38, 39, 57, 275, 276
- implements keyword, 144
- import keyword, 6–9
- import statement, 7
- importing nested classes, 164
- imports, static, 85–86
- inclusive OR operator (|), 44, 53, 54
- increment operators (++), 44, 49–50
- indeterminate output, of multithreaded programs, 346, 348, 367, 369
- information hiding, 384–385. *See also* tight encapsulation
- inheritance, 393–397. *See also* access modifiers; polymorphism
  - “is a” relationship and, 390, 391, 392, 409
  - modifiers and
    - abstract, 397–399
    - access, 393–397, 410
    - final, 399–401
  - multiple, with interfaces, 147
  - polymorphism and, 402
- initialization, 100. *See also* constructors; instantiation process
  - explicit, 81–83
  - of instance variables, 80–81
  - of local variables, 86–87
  - of variables, 78
- initializers. *See* array initializers; instance initializers; static initializers
- Inner class, 153, 154
- inner classes, 152
  - anonymous, 159–162
  - behind-the-scene details of, 155
  - as event handlers, 157, 161–162
  - local, 152, 158–159
  - member, 150–157
  - nested, 156
  - reuse and, 157, 162
- input streams, 282, 282
- input/output
  - binary data, 281. *See also* streams
  - character/string data, 281. *See also* reader classes; writer classes
  - files, 285–301, 324
- InputStream, 281, 282, 284
- InputStreamReader, 282, 284
- insert method, 275, 279, 280, 281, 325
- instance initializers, 101, 111–114
  - curly braces and, 112
  - defined, 97, 111, 165
  - purpose of, 114
  - syntax for, 112
- instance methods, 121–124
  - behind-the-scene details of, 124
  - defined, 121
  - overridden, 137
  - references and, 122
- instance variables (fields), 80–83, 97, 165
  - explicit initialization of, 81–83
  - initialization of, 80–81
  - lifetime of, 81
  - static variables *v.*, 166
- instanceof operator, 44, 51–52, 244, 406–407, 411
  - as Boolean operator, 406
  - ClassCastException and, 244, 406
  - polymorphism and, 406–407
- instantiation process, 100–102, 165
- int, 22
- Integer (wrapper class), 192, 194, 270, 463
- integer\_variable, 192
- interface(s), 143–147. *See also specific interfaces*
  - anonymous inner classes and, 161
  - API, 4, 5

- interface(s) (*continued*)
- classes *v.*, 143
  - collections, 426–432, 476
  - Comparable, 432–433
  - data types and, 146
  - declaring, 143–147
  - defined, 143
  - Drawable, 144, 145, 146, 147
  - extending, 147
  - generic, 451–452
  - GUI, 17, 100
  - implementing, 144–146
    - polymorphism and, 402
  - List, 426, 428, 438, 440, 455, 466
  - Map, 426, 427, 431, 445, 446, 475, 488
  - multiple inheritance with, 147
  - properties of, 143
  - public and, 143
  - purpose of, 145
  - Queue/Deque, 426, 428, 430, 475
  - Runnable, 344–346, 368, 369
  - Serializable, 301–303
  - Set, 426, 428, 429, 441
- internal invariants, 221
- InterruptedException, 116, 232, 237, 353
- intValue method, 271
- invalid identifiers, 79
- InvalidLogger class, 400
- InvalidShape class, 397, 398
- invariants
  - class, 221, 223
  - control flow, 221, 222
  - internal, 221
- IOException, 8, 231, 232, 233, 237, 239, 266, 267
- is-a relationship, 390–392, 409, 410
- isValid method, 223, 224
- Iterable, 201, 202, 203
- Iterator `<E>` descendingIterator(), 444
- Iterator `<E>` iterator(), 444
- iterator method, 427, 443, 444. *See also* collections; generics
- iterators, 197, 201, 202, 204, 206, 209, 211, 215. *See also* break statements; continue statements; repetition control structures
- 
- ## J
- JAR files, 17–18
- Java
  - API. *See* API
  - applications, command line and, 12–22
  - Collections Framework. *See* Collections Framework
  - fundamentals, 1–75
  - modifiers. *See* abstract modifier; access modifiers; final modifier
  - naming conventions. *See* naming conventions
  - OOP and, 2. *See also* object-oriented programming
- Java Platform, Standard Edition (JavaSE), 4, 5, 32
- Java Runtime Environment (JRE), 2, 11
- .java source file, 97, 165
- JavaBeans, 118
  - event methods, 120–121
  - naming convention, 118–121, 165
  - read/write properties, 120
- java.io package, 281
  - file input/output and, 285–301, 324
  - high-level streams in, 283–285, 284, 324
  - input streams of, 282, 282
  - low-level streams in, 283–285, 284, 324
  - object serialization and, 301–306, 324
  - output streams of, 282, 282
  - reader classes of, 281–283, 282, 324
  - writer classes of, 281–283, 283, 324
- java.lang package, 270. *See also* strings; Throwable types; wrapper classes

strings and, 274  
 Throwable types and, 242  
 wrapper classes and, 270

JavaSE (Java Platform, Standard Edition),  
 4, 5, 32

java.text package, 270, 306  
 formatting/parsing data and, 306–315,  
 324, 325

java.util package. *See also* collections  
 interfaces  
 Collections API of, 89, 203, 270, 302  
 collections interfaces in, 426–432, 476

java.util.regex package, 269, 315. *See also*  
 regular expressions

join method, 354, 357, 361, 368, 369

JRE. *See* Java Runtime Environment

july array, 90, 91, 92

---

## K

K (map key), 452

keyboard input, Scanner class and,  
 323, 325

keyword(s)  
 abstract, 138, 140  
 counting, maps and, 431–432  
 enum, 148, 166, 167  
 extends, 147, 455, 456  
 final, 137  
 implements, 144  
 import, 6–9  
 native, 117, 165  
 new operator, 29, 80, 82, 100, 101,  
 153, 154  
 package, 5–6  
 super  
 in constructors, 108–111, 166  
 default constructors and, 110–111  
 parent class constructors and, 108,  
 110, 165, 166  
 as reference, 108, 134

synchronized, 353, 358, 362, 363,  
 368, 369

this  
 in constructors, 105–107, 166  
 as reference, 105, 134

throws, 233

tokens as, 79

transient, 301, 302

keyword counters, 431–432

keywordFound method, 432

---

## L

labeled break statements, 213

labeled continue statements, 217, 218

labels, 213, 214, 217, 218. *See also* break  
 statements; continue statements

language codes (online), 308

Lassie object, 35, 60

last-in, first-out (LIFO), 428, 441.  
*See also* Stack

leftClick method, 87, 88

length attribute, 243

less than (<), 50–51

less than or equal (<=), 50–51

lifetime, of instance variables, 81

LIFO (last-in, first-out), 428, 441.  
*See also* FIFO; Stack

line numbers, 3

LineNumberReader, 284

LinkedHashMap, 431, 447, 475

LinkedHashSet, 429, 429, 441, 443, 475

LinkedList, 428, 430, 440, 475, 476

Lion class, 132, 133, 134, 137, 138

List interface, 426, 428, 438, 440, 455, 466

lists, 427–428, 476. *See also* ArrayList;  
 LinkedList; Stack; Vector  
 classes, 428  
 conversions  
 arrays into lists, 472  
 lists into arrays, 466–467

- lists (*continued*)
    - defined, 426, 427
    - generics and, 438–441
    - searching, 467–469
    - sorting, 461–467, 476, 477
  - literals, tokens as, 79. *See also* string literals
  - local inner classes, 152, 158–159
    - anonymous inner classes *v.*, 161
  - local variables, 86–88, 165
  - Locale class, 308
  - locales, 308, 325
  - locks. *See* monitor locks
  - logErrors method, 126, 127
  - logical operators, 44, 53–55
    - AND (&&), 44, 53, 54
    - OR (| |), 44, 53, 54
  - long, 22
  - Long class, 270, 463
  - loops. *See* do-while loops; enhanced for loops; for loops; while loops
  - loose coupling, 385–388, 409. *See also* high cohesion; tight encapsulation
    - high cohesion and, 389
    - tight encapsulation and, 386
  - lower-bound wildcards, 459–460
  - low-level streams, 283–285, 284, 324
- 
- M**
- main method, 188
  - Main program, 347, 348
  - main threads, 342
  - Mammal class, 132, 133, 134, 138, 139, 141, 142
  - Mammal interface, 402, 403, 404
  - “many forms.” *See* polymorphism
  - maps, 430–432, 476. *See also* HashMap; LinkedHashMap; TreeMap
    - arrays *v.*, 445
    - classes, 431
    - defined, 426, 430, 445
    - generics and, 445–449
    - keyword counting and, 431–432
    - phonebook application and, 430, 446, 447
  - Map interface, 426, 427, 431, 431, 445, 446, 475, 488
  - map key (K), 452
  - map value (V), 453
  - Map.Entry <K, V> ceilingEntry(K key), 448
  - Map.Entry <K, V> floorEntry(K key), 448
  - Matcher class, 315–316, 325
  - matcher method, 315, 325
  - Math class, 124
  - member inner classes, 150–157
  - memory
    - heap, 28
    - primitive types in, 23, 23
    - reference types in, 24, 24
  - metacharacters, 316, 316–318. *See also* regular expressions
    - ? (question mark), 269, 315, 316, 318
    - character classes, 319–320
    - \d, 269, 316, 319, 320
    - dot (.), 269, 315, 316, 317
    - parentheses ( ), 269, 316, 319
    - plus (+), 269, 315, 316, 318
    - \s, 269, 316, 319, 320
    - square brackets ([ ]), 269, 315, 316, 317
    - star (\*), 269, 315, 316, 317, 318
    - \w, 269, 316, 319, 320
  - methods, 116–138. *See also specific methods*
    - abstract, 140–143
    - class. *See* static methods
    - declaring, 116–138, 165
    - defined, 97
    - enum, declaring, 150–151
    - final, 137–138, 410
    - generic, 453–455
    - generic types and. *See* generic types
    - instance, 121–124

- naming convention, 117, 118–121
  - polymorphic parameters of, 407–408
  - private, method overriding and, 132
  - static, 124–126
    - hidden, 137
  - synchronized, 117, 362–363
  - virtual, 404–405, 411
  - method declarations, 116–118, 165
    - elements in, 116
    - syntax, 116
  - method hiding, 135–137
    - method overriding *v.*, 136, 137
    - non-final static methods and, 132, 401
  - method overloading, 128–131, 165
    - data type promotion and, 129–130
    - method overriding *v.*, 166
  - method overriding, 131–134, 165
    - abstract methods and, 410
    - final methods and, 410
    - method hiding *v.*, 136, 137
    - method overloading *v.*, 166
    - OOP and, 131
    - private methods and, 132
  - method parameters, validating, 224
  - method signatures, 118
  - mkdir method, 287, 337
  - modifiers. *See* abstract modifier; access
    - modifiers; final modifier
  - modifyStacks, 40, 41
  - modulus operator (%), 44, 48–49
  - monitor locks
    - deadlock and, 360–361
    - features of, 358
    - synchronized blocks and, 358–361
    - synchronized methods and, 362–363
  - months array, 95
  - MountainLion class, 138
  - Mouse class, 87
  - Movie class, 82
  - multidimensional arrays, 93–95
  - multiple catch clauses, 229–231
  - multiple classes in single file, 4, 62
  - multiple inheritance, with interfaces, 147
  - multiplication (\*) operator, 48
  - multiplicative operators, 48–49
    - division (/), 48
    - modulus (%), 44, 48–49
    - multiplication (\*), 48
  - multithreaded programs
    - indeterminate output of, 346, 348, 367, 369
    - thread synchronization and, 355–357
  - mutator methods (setter methods), 119, 165, 382, 387, 423
  - MyButtonListener class, 157
  - MyErrorLog class, 126, 127
  - MyFileReader class, 229–230
  - MyLogger class, 400
  - myloop label, 213, 214
  - MyMath class, 457
  - MyNewLogger class, 401
  - MyNumberFormatter class, 115
  - MyStack class, 355, 356, 357, 362
  - MyStack2 class, 362, 363, 364, 365, 366
  - MyStaticLogger class, 400, 401
  - MyThread class, 347, 348
- 
- N**
- N (number), 453
  - names array, 95, 95
  - namespaces, packages and, 9
  - naming conventions
    - for constants, 400
    - for final variables, 400
    - for generics, 452–453
    - JavaBeans, 118–121, 165
    - for methods, 117, 118–121
    - online information, 400
    - for packages, 9
  - native keyword, 117, 165
  - natural ordering, 433, 477. *See also*
    - compareTo; sorting arrays; sorting lists

natural ordering (*continued*)  
 comparators *v.*, 477  
 compareTo and, 433, 435, 461, 462  
 of numeric classes, 463  
 of strings, 433, 463–464  
   lexicographical and, 433, 463, 475  
 of wrapper classes, 463–464  
 NavigableMap interface, **448–449**. *See also* TreeMap  
 NavigableMap  $\langle K, V \rangle$  descendingMap(), 448  
 NavigableMap  $\langle K, V \rangle$  subMap(), 448  
 NavigableSet  $\langle E \rangle$  subSet(), 444  
 NavigableSet interface, **444–445**. *See also* TreeSet  
 NavigableSet  $\langle K \rangle$  descendingKeySet(), 448  
 nested classes, **152–164**, 166, 167. *See also*  
   inner classes  
   defined, 97  
   importing, 164  
   member inner classes, 150–157  
   static, 152, 162–164  
 nested enhanced for loops, 205  
 nested for loops, 199–200  
 nested inner classes, 156  
 new operator (keyword), 29, 80, 82, 100, 101, 153, 154  
 new threads, 349–351, 368, 369  
 “next” methods, 322, 323  
 NoClassDefFoundError, 14, 16, 73, 242, **249**  
 NonFictionBook class, 108, 109  
 non-final static methods, method hiding and, 132, 401  
 nongeneric collections, 436–437  
 not equal to (!=), 56–58  
 notify method, 353, 354, 354, 363–369  
 notifyAll method, 353, 354, 354, 363–369  
 null type, 26  
 NullPointerException, 122, 194, 229, 232, 233, 237, 241, 242, **246**, 247, 267, 268, 337

numbers  
 (N), 453  
 strings converted into, 310–311  
 Number class, 457  
 NumberFormat class, **306–309**, 324  
   DecimalFormat (child class), 309–310  
   format methods, 307–309  
   parse method, 310–312, 324  
   static methods, 306–307  
 NumberFormatException, 242, **246–247**, 266, 267, 272  
 numeric types. *See also* boolean;  
   NumberFormatException; primitive types;  
   *specific numeric types*  
 boolean expressions and, 51  
 increment/decrement operators and, 49  
 NumberFormatException and, 242  
 numeric wrapper classes. *See* wrapper classes

---

## O

objects. *See also* classes; collections; methods; polymorphism  
 array, 90–91  
 cookie analogy and, 2  
 defined, 2  
 equality of, 58–61  
 heterogeneous collections of, 409  
 house analogy and, 2, 85  
 initialization. *See* initialization  
 instantiation process, 100–102, 165  
 new operator and, 29, 80, 82, 100, 101, 153, 154  
 passing, 41  
 reference types *v.*, 29, 41  
 serialization. *See* serialization  
 Object class, 58, 60  
   arrays and, 165  
   ClassCastException and, 244  
   generics as, 451

- objects and, 52, 408
- as parameters, 408
- polymorphism and, 408
- primitive types and, 131, 135
- toString method and, 276
- ObjectInputStream class, 282, 284, 301, 304–306, 324
- ObjectOutputStream class, 303–304, 324
- object-oriented programming (OOP), 2, 381–424. *See also* access modifiers; classes; encapsulation; inheritance; OO design principles; polymorphism; reuse
- abstraction in, 139
- design relationships, 390–393, 409
- instanceof operator. *See* instanceof operator
- Java and, 2
- method overriding and, 131. *See also* method overriding
- review questions/answers, 412–424
- static fields and, 85
- OO design principles, 382, 409. *See also* high cohesion; loose coupling; tight encapsulation
- goals of, 385, 388, 393
- has-a relationship and, 392–393, 409
- high cohesion and, 388
- is-a relationship and, 390–392, 409
- loose coupling and, 382, 385, 386, 387, 388
- tight encapsulation and, 382, 385, 386
- OOP. *See* object-oriented programming
- operators, 43–58, 62. *See also specific operators*
- arithmetic, 46–50
  - additive, 44, 46–47
  - decrement (--), 44, 49–50
  - increment (++), 44, 49–50
  - multiplicative, 48–49
- assignment, 44, 44–46
- bitwise, 44, 53–55
- comparison (=), 56–58, 62, 274, 433–435, 477
- conditional (ternary) (? :), 44, 55
- equality, 56–58
  - comparison (=), 56–58, 62, 274, 433–435, 477
- instanceof, 44, 51–52, 244
- list of, 44
- logical, 44, 53–55
  - AND (&&), 44, 53, 54
  - OR (||), 44, 53, 54
- multiplicative, 48–49
  - division (/), 48
  - modulus (%), 44, 48–49
  - multiplication (\*), 48
- new. *See* new operator
- precedence of, 43, 44
- relational, 44, 50–51
  - greater than (>), 51
  - greater than or equal (>=), 51
  - less than (<), 50–51
  - less than or equal (<=), 50–51
- shift, 44
- tokens as, 79
- unary, 44
- OR operators
  - exclusive (^), 44, 53, 54
  - inclusive (|), 44, 53, 54
  - logical (||), 44, 53, 54
- order of precedence, 43, 44
- ordering. *See* natural ordering
- ordinal method, 150, 184
- outcomes, assertions and, 222
- Outer class, 153, 154
- output, code, 61. *See also* input/output
- output streams, 282, 282
- OutputStream, 281, 282, 284
- OutputStreamWriter, 283, 284
- overloading. *See* method overloading
- overridden instance methods, 137
- overriding. *See* method overriding

---

**P**

- packages, 4–11. *See also* java.io package;  
java.lang package; java.text package;  
java.util package; java.util.regex  
package
  - benefits of, 4
  - classes in, 5–6, 61
  - directory structure, 9–10
  - names, JAR files and, 18
  - namespaces and, 9
  - naming convention for, 9
  - online information, 5
  - as tabbed folders, 5
  - unnamed, 6
- package keyword, 5–6
- package-level access. *See* default access
- parameter lists, 117, 165
- parameters, 36
  - method, validating, 224
  - Object class and, 408
  - polymorphic, 407–408
  - type
    - bounded, 455
    - formal, 450, 451, 456
- parent class. *See also* child classes
  - construction, 100–102, 165
  - constructors
    - instance initializers and, 111
    - super keyword and, 108, 110,  
165, 166
  - is-a relationship and, 390, 391,  
409, 410
  - method hiding and, 135, 136, 166
  - method overloading and, 128, 129, 166
  - method overriding and, 165, 166
  - polymorphism and, 402, 403, 407, 410,  
411
- parentheses (metacharacter), 269, 316, 319
- parse methods
  - DateFormat class, 314–315
  - NumberFormat class, 310–312, 324
  - currency and, 312
  - strings into numbers and, 310–311
- ParseException, 310, 311, 314, 315
- parsing
  - currency, 312
  - data, 306–315, 324, 325
  - strings into numbers, 310–311
  - strings into primitive types, 272
- passing
  - arguments *by value*, 36
  - objects, 41
  - primitive types, 37
  - reference types, 37, 41
- Pattern class, 315–316, 325
  - character classes, 319–320, 320
- Payroll class, 388, 389
- Person class, 391, 392, 393
- Pet class, 390, 391, 402, 403, 404, 409
- Phone class, 394, 395, 396
- phonebook application, 430, 446, 447. *See also* maps
- Picture class, 144, 145, 146
- PipedInputStream, 282, 284
- PipedOutputStream, 282, 284
- Plant interface, 146
- Platypus class, 139, 140
- plus (+)
  - increment operators (++), 44, 49–50
  - metacharacter, 269, 315, 316, 318
  - operator, string concatenation and, 46,  
47, 276
- pointer, 23
- polymorphic parameters, 407–408
- polymorphic references, casting, 47, 405–409
- polymorphism, 401–409, 410, 411
  - child classes and, 402, 403, 410, 411
  - generics and, 456, 458, 476
  - heterogeneous collections and, 409
  - inheritance and, 402
  - instanceof operator and, 406–407
  - interface implementation and, 402
  - is-a relationship and, 392

- Object class and, 408
  - parent class and, 402, 403, 407, 410, 411
  - virtual method invocation and, 404–405
  - pop method. *See also* push method; Stack
    - Stack class and, 39, 440
    - synchronized methods and, 355, 356, 362, 363
  - post conditions, 221
  - pound sign (#), 309, 310
  - precedence, of operators, 43, 44
  - precision, of doubles, 159
  - preconditions, 221
  - preemptive scheduling, 351, 355, 360, 368
  - Preference class, 408
  - primitive types, 22, 22–23. *See also* data types; numeric types; reference types; *specific primitive types*
    - autoboxing of, 131, 271, 272–274, 325
    - casting, 47
    - parsing strings into, 270, 322
    - /reference types, passing, 37
    - unboxing of, 271, 272–274, 325
  - primitive wrapper classes. *See* wrapper classes
  - print method, 295
  - printf method, 295–298, 324
  - println method, 295
  - printStackTrace methods, 228
  - PrintWriter class, 294–295
    - format/printf methods, 295–298, 324
    - Formatter class *v.*, 298
  - priority, thread, 343, 351, 352, 353, 368
  - PriorityQueue, 430, 475
  - private (access modifier), 393, 410. *See also* access modifiers
    - constructors and, 102
    - member inner class and, 153
    - methods and, 116
    - top-level classes and, 3, 4
  - private methods
    - abstract methods and, 398
    - method overriding and, 132
  - processes, 342. *See also* threads
  - producer/consumer model, 364–367, 368, 369
  - Product objects, 441, 442, 443, 444, 468, 469
  - ProductComparator class, 469
  - protected (access modifier), 393, 410. *See also* access modifiers
    - constructors and, 102
    - member inner class and, 153
    - methods and, 116
    - top-level classes and, 3, 4
  - public (access modifier), 393, 410. *See also* access modifiers
    - constructors and, 102
    - interfaces and, 143
    - member inner class and, 153
    - methods and, 116
    - top-level classes and, 3, 4
  - public boolean add(E e), 427
  - public boolean contains(Object e), 427
  - public boolean remove(Object e), 427
  - public Iterator <E> iterator(), 427
  - push method. *See also* pop
    - method; Stack
    - Stack class and, 39, 440
    - synchronized methods and, 355, 356, 362, 363
  - PushbackInputStream, 282, 284
  - PushbackReader, 282, 284
  - Pusher class, 356
  - Pusher2 class, 363
- 
- Q**
- question mark (?). *See also* metacharacters; wildcard generic types
    - metacharacter (?), 269, 315, 316, 318
    - wildcard generic types (? , ? extends, ? super), 458

Queue interface, 426, 428, 430, 475  
 queues, 429–430, 476. *See also*  
   ArrayDeque; LinkedList;  
   PriorityQueue  
 classes, 430  
 defined, 426  
 dequeues and, 430, 475  
 FIFO and, 426, 429, 430

---

## R

ReadAFile class, 350  
 reader classes, 281–283, 282, 324. *See also*  
   streams  
 readLine method  
   BufferedReader, 285  
   Console class, 299  
 readObject method, 304  
 readPassword method, 299, 300  
 Rectangle class, 223, 224  
 reference casting, 47, 405–409  
 reference types, 23–28. *See also* arrays;  
   data types; primitive types; strings;  
   *specific reference types*  
   instance methods and, 122  
   objects *v.*, 29, 41  
   polymorphic, casting of, 47,  
     405–409  
   /primitive types, passing, 37  
   switch statement and, 194  
 ReferenceDemo program, 57, 58  
 regular expressions, 315–322, 325  
   character classes in, 319–320, 320  
   Matcher class and, 315–316, 325  
   metacharacters of, 316, 316–318  
   parentheses in, 269, 316, 319  
   Pattern class and, 315–316, 325  
   String.split method and, 320–322, 325  
 relational operators, 44, 50–51  
   greater than (>), 51  
   greater than or equal (>=), 51

  less than (<), 50–51  
   less than or equal (<=), 50–51  
 repetition control structures, 188, 249,  
   250. *See also* do-while loops; enhanced  
   for loops; for loops; while loops  
   continue statements and, 215–218  
 replace(), 280  
 ReturnDemo program, 43  
 reuse. *See also* high cohesion; loose  
   coupling; OO design principles;  
   polymorphism; tight encapsulation  
   high cohesion and, 388, 389  
   inner classes and, 157, 162  
   is-a/ has-a relationships and, 393  
   loose coupling and, 389, 423  
   string literals and, 57  
   tight encapsulation and, 423  
 reverse(), 280  
 Reverse class, 465, 466  
 RotaryPhone class, 395, 396  
 rounding, 307  
 run method, 160, 343, 344, 345, 346, 347,  
   349, 350, 352, 355, 368  
 Runnable interface, 344–346, 368, 369  
 runnable target, 343, 344, 347  
 runnable threads, 349, 351–353, 368, 369  
 running Java applications (from command  
   line), 12–22  
   ColorChanger and, 14, 15, 16, 17  
   TestColors and, 15, 16, 17  
 runtime exceptions, 231, 232, 233,  
   237. *See also* ArithmeticException;  
   NullPointerException  
 RuntimeException, 231, 232, 232, 267

---

## S

%s, 297, 325  
 \s (metacharacter), 269, 316, 319, 320  
 SayHello class, 13, 14, 344, 345, 346, 352  
 Scanner class, 322–323, 325

- constructors in, 322
- keyboard input and, 323, 325
- scope, 80, 165. *See also* variables
  - of do-while loop variables, 211
  - of enhanced for loop variables, 202–203
  - of for loop variables, 199
- ScoreKeeper class, 271, 272, 273
- search engines, keyword counters and, 431–432
- searching arrays, 473–475, 477
- searching lists, 461, 467–469
- Season enum, 148
- send methods, 129, 130, 131
- separators, tokens as, 79
- Serializable interface, 301–303
- serialization, 301–306, 324, 325
- SerializeDemo program, 303, 304
- sets, 428–429, 476. *See also* HashSet;
  - LinkedHashSet; TreeSet
  - classes, 429
  - defined, 426, 428
  - generics and, 441–445
- Set interfaces, 426, 428, 429, 441
- set methods, 119, 120
- setenv command, 11
- setPriority method, 351
- setter methods (mutator methods), 119, 165, 382, 387, 423
- Shape class, 397
- shift operators, 44
- Shipment class, 163, 164
- Short, 192, 194, 270, 271, 463
- short, 22
- short-circuit behavior, 54, 55
- showExceptions method, 460
- simple assignment operator, 44, 45
- sleep method, 116, 352, 354, 368, 405
- Snoopy object, 35, 36
- sort methods
  - for arrays, 470
  - for lists, 461–462, 466–467
- sorting arrays, 470–473, 477
- sorting lists, 461–467, 476, 477
  - comparable sorting, 462–464
  - comparator sorting, 465–467
- source code
  - /bytecode, separating, 19
  - line numbers and, 3
- specifiers. *See* access modifiers
- split method. *See* String.split method
- square brackets ([ ])
  - arrays and, 21, 89, 93, 294
  - metacharacter, 269, 315, 316, 317
- Square class, 398
- Stack, 39, 428, 440–441, 475, 476
- Stack objects, 39, 40, 41
- StackDemo, 37, 39, 41
- StackOverflowError, 242, 248
- star (\*), 269, 315, 316, 317, 318
- start method, 160, 345, 348, 349, 350, 369
- starting threads, 351, 355, 360, 361, 369
- statements. *See* decision making control
  - structures; repetition control
  - structures; *specific statements*
- states of threads. *See* thread states
- States program, 285, 286
- static fields (class variables, static variables), 83–86, 97, 165
  - global variables and, 83, 85
  - instance variables *v.*, 166
  - OOP and, 85
  - static imports, 85–86
- static imports, 85–86
- static initializers, 114–115
  - class loader and, 115
  - defined, 97, 114
  - MyNumberFormatter class and, 115
- static methods (class methods), 124–126.
  - See also specific static methods*
  - global methods *v.*, 125
  - hidden, 137
  - non-final, method hiding and, 132, 401
  - NumberFormat class, 306–307

- static nested classes, 152, **162–164**
  - importing, 164
  - top-level classes *v.*, 163
- static variables. *See* static fields
- StaticProblem class, 125
- stop method (deprecated), 348
- streams, **281–285**. *See also* chaining streams
  - byte. *See* FileInputStream class; FileOutputStream class
  - character. *See* FileReader class; FileWriter class
  - high-level, 283–285, 284, 324
  - input, 282, 282
  - low-level, 283–285, 284, 324
  - output, 282, 282
  - readers/writers *v.*, 281–283, 324
- strings, **274–281**
  - comparing, equals method and, 276
  - into dates, parse method and, 314–315
  - natural ordering of, 433, 463–464
    - lexicographical and, 433, 463, 475
  - into numbers, parse method and, 310–311
  - into primitive types, 270, 322
  - into tokens, Scanner class and, 322, 323
- String class, 275–278, 324
  - equals method and, 275
  - StringBuilder *v.* StringBuffer *v.*, 275
- string concatenation
  - + operator and, 46, 47, 276
  - optimizing, 277
- string literals, 25, 57, 275, 277, 316, 320. *See also* regular expressions
  - reuse and, 57
  - string pool and, 25, 275
- string manipulation methods
  - (StringBuilder/StringBuffer classes), 280
- string methods. *See* append method; concat method; insert method
- String objects, immutable quality of, 25, 38, 39, 57, 275, 276
- string pool, 25, 275
- StringBuffer/StringBuilder classes, 275, **278–281**, 324
  - append method, 275, 278, 279, 280, 281, 288 325
  - insert method, 275, 279, 280, 281, 325
  - string manipulation methods, 280
  - String *v.*, 275
- StringReader, 282, 284
- String.split method, **320–322**, 325. *See also* regular expressions
- StringWriter, 283, 284
- Student1 class, 382
- Student2 class, 383, 385
- subclasses, concrete, 141
- super keyword
  - in constructors, 108–111, 166
  - default constructors and, 110–111
  - parent class constructors and, 108, 110, 165, 166
  - as reference, 108, 134
- super.finalize, 33, 33
- supertypes, generic, 459, **461**
- switch statements, **192–197**
  - enums and, 149, 194–196
  - if statements *v.*, 192
  - reference types and, 194
  - rules for, 192–193
  - syntax, 192
- synchronization. *See* thread synchronization
- synchronized blocks, 358–361
- synchronized keyword, 353, 358, 362, 363, 368, 369
- synchronized methods, 117, **362–363**
- syntax
  - basic for loops, 198
  - break statements, 212
  - continue statements, 216
  - do-while loops (do statements), 209

- enhanced for loops (for-each loops), 201
- finally block, 238
- generic methods, 454
- generics, 450, 454, 455, 476
- if statements (if-else, if-then-else), 189
- instance initializers, 112
- method declarations, 116
- switch statements, 192
- system threads, 342
- System.gc method, 32–33
- System.in
  - Console class and, 298, 301
  - Scanner class and, 323
  - static fields and, 85
- System.out
  - Console class and, 298, 301
  - static fields and, 85

---

## T

- T (generic data type), 453
- tabbed folders, packages as, 5
- tailMap method, 448
- tailSet method, 444
- Television class, 80, 81
- terminated threads, 349, 355, 368, 369
- ternary operator. *See* conditional operator
- TestColors class, 221, 222
- TestColors program, 15, 16, 17, 18
- this keyword
  - in constructors, 105–107, 166
  - as reference, 105, 134
- threads
  - blocked, 349, 353, 368, 369
  - components of, 343
  - concurrency and, 342
  - defined, 342
  - IllegalThreadStateException and, 351, 355, 369
  - join method, 354, 357, 361, 368, 369
  - main, 342
  - multithreaded programs
    - indeterminate output of, 346, 348, 367, 369
    - thread synchronization and, 355–357
  - MyThread class and, 347, 348
  - new, 349–351, 368, 369
  - overview of, 342
  - preemptive scheduling and, 351, 355, 360, 368
  - run method, 160, 343, 344, 345, 346, 347, 349, 350, 352, 355, 368
  - runnable, 349, 351–353, 368, 369
  - sleep method, 116, 352, 354, 368
  - starting, 351, 355, 360, 361, 369
  - stopping, 348
  - system, 342
  - terminated, 349, 355, 368, 369
  - timed-waiting, 349, 353–354, 368, 369
  - user-defined, 342
  - waiting, 349, 353–354, 368, 369
  - writing, 343–348, 368, 369
    - extending Thread class and, 343, 346–348, 368, 369
    - Runnable interface and, 344–346, 368, 369
    - yield method, 352, 353, 355, 360, 368
- thread priority, 343, 351, 352, 353, 368
- thread states, 349–355, 368, 369
- transitions
  - new to runnable, 350, 369
  - runnable to blocked, 353, 369
  - runnable to terminated, 355, 369
  - sleep/yield methods and, 352
  - wait/notify methods and, 354
- thread synchronization, 355–368
  - monitor locks
    - deadlock and, 360–361
    - features of, 358
    - synchronized blocks and, 358–361
    - synchronized methods and, 362–363
  - multithreaded program and, 355–357

- ThreadGroup class, 344
- Thread.State, 349
- Throwable class, 228–229
- Throwable types, 232, 242–249. *See also* exceptions
- throws clause, 118, 165, 232
- throws keyword, 233
- tight encapsulation, 382–385, 409, 423.  
*See also* high cohesion; loose coupling  
 high cohesion and, 389  
 loose coupling and, 386
- timed-waiting threads, 349, 353–354, 368, 369
- toArray method, 466–467
- tokens. *See also* identifiers; keywords; literals; operators; Scanner class; separators  
 parsing strings into, 322, 323  
 types of, 79
- Tomato class, 104, 105
- Tooth Fairy string, 276, 277
- top-level classes  
 access modifiers for, 4  
 multiple, in single file, 4, 62  
 private/protected and, 3, 4  
 static nested classes *v.*, 163
- toString method, 149, 151, 229, 276, 294, 305, 306, 357
- transient keyword, 301, 302
- transitions, of thread states  
 new to runnable, 350, 369  
 runnable to blocked, 353, 369  
 runnable to terminated, 355, 369  
 sleep/yield methods and, 352  
 wait/notify methods and, 354
- TreeMap, 431, 446–447, 475, 476  
 NavigableMap interface and, 448–449
- TreeSet, 429, 443–445, 475, 476
- try statements, 227–229  
 catch clauses and, 227, 229–231  
 curly braces and, 227  
 finally block and, 238–241, 250  
 syntax, 227
- try-catch-finally block, 238–241, 250
- try-finally statement, 240–241
- types. *See* data types; generic types; numeric types; primitive types; reference types; Throwable types
- type erasure, 451
- type parameters  
 bounded, 455  
 formal, 450, 451, 456
- 
- ## U
- unnamed packages, 6
- unary operators, 44
- unbounded wildcards, 458
- unboxing, 271, 272–274, 325
- underscore character (`_`), 78, 183  
`\w` metacharacter and, 316
- UNICODE format, 23, 78
- unreachable code, while loops and, 208
- upper-bound wildcards, 458–459
- useDelimiter methods, 322, 323
- user-defined threads, 342
- 
- ## V
- V (map value), 453
- Vacation class, 212, 213
- vaccinate method, 407
- valid identifiers, 79
- validating method parameters, 224
- value types. *See* primitive types
- valueOf method, 150, 247
- values array, 94
- values method, 149
- variables, 78–79. *See also specific variables*  
 class. *See* static fields  
 declaring, 78–79, 165  
 defined, 78

final, 196–197  
 global, 83, 85. *See also* static fields  
 identifiers and, 78–79, 79  
 initialization of, 78  
 instance, 80–83, 97, 165  
 local, 86–88, 165  
 scope, 80, 165  
 static. *See* static fields  
 variable-length argument list (. . .), 12, 118, 126, 165  
 variable-length arguments, 126–128  
 Vector, 34, 35, 82, 428, 440, 475, 476  
 Vehicle class, 112  
 Vet class, 407  
 virtual method invocation, 404–405, 411.  
     *See also* polymorphism  
 virtual methods, 404–405, 411

---

## W

\w (metacharacter), 269, 316, 319, 320  
 wait method, 353, 354, 354, 363–369  
 waiting threads, 349, 353–354, 368, 369  
 walk method, 141, 142, 143  
 while loops, 206–208  
     curly braces and, 206  
     do statements *v.*, 208  
     examples, 206, 207  
     rules for, 206  
     syntax, 206  
     unreachable code and, 208  
 whitespace, 322, 323  
     \s, 269, 316, 319, 320

wildcard (\*), 7, 8, 85. *See also*  
     metacharacters  
 wildcard generic types, 458–460  
     lower-bound, 459–460  
     unbounded, 458  
     upper-bound, 458–459  
 wrap method, 454, 455  
 wrapper classes (numeric classes),  
     270–272. *See also* autoboxing;  
     primitive types  
     Boolean, 270  
     Byte, 192, 194, 270, 463  
     Character, 192, 194, 270  
     Double, 270, 272, 273, 312, 339, 463  
     Float, 270, 271, 272, 273, 452, 463  
     Integer, 192, 194, 270, 463  
     Long, 270, 463  
     natural order of, 463–464  
     parsing strings with, 272  
     properties of, 270–271  
     Short, 192, 194, 270, 271, 463  
 writeObject method, 408  
 writer classes, 281–283, 283, 324. *See also*  
     streams

---

## X

XBOX, 195, 196

---

## Y

yield method, 352, 353, 355, 360, 368