



Index

COPYRIGHTED MATERIAL

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

A

abstract classes
 concrete classes from, 262–263
 creating, 259
 defining, 260–262
 extending, 263–265
abstract specifiers
 interfaces, 267–269
 methods, 168, 271–273
access modifiers, 173
 default, 175
 description, 7
 inheritance, 237
 methods, 166–167
 private, 173, 174
 protected, 176, 176
 public, 180–181
 static, 181–188
add() method, 130–131
addition
 dates, 142–143
 precedence, 53–54
ampersands (&) for logical operators, 64, 64
ancestors in inheritance, 234
“and” logical operator, 64, 64
anonymous arrays, 120
append() method, 115
args parameters in main(), 7–8
arithmetic operators, 53–55
ArithmeticException class, 314
ArrayIndexOutOfBoundsException class, 123, 314–315
ArrayLists, 129
 arrays from, 136–137
 autoboxing, 136
 creating, 129
 methods, 130–134
 sorting, 138
 wrapper classes, 134–135
arrays, 119
 ArrayLists, 129
 declaring, 121
 indexes, 7, 120, 120

 multidimensional, 126–129, 127–128
 parameters, 7
 primitives, 119–121, 119–120
 reference variables, 121, 122
 searching, 125–126
 sorting, 124–125
 varargs, 126
 working with, 123–124
arrow operator (->) for lambda expressions, 211–212, 212
asList() method, 187–188
assignment operators
 compound, 62–63
 overview, 60
 precedence, 53
assumed keywords in interfaces, 268
asterisks (*)
 comments, 4–5
 packages, 10
asymmetric arrays, 128, 128
autoboxing
 overloading methods, 193
 type conversions, 136

B

B prefix for binary, 22
backed lists, 137
base 10 numbering system, 22
Beginning Java forum, 6
binary number format, 22–23
binary operators
 arithmetic, 53–55
 assignment, 60
 casting, 60–61
 compound assignment operators, 62–63
 equality, 65–66
 logical, 64–65, 64
 numeric promotion, 55–57
 relational, 63
binary searches, 125–126
bitwise operators, 64, 64

blocks
 description, 67
 initialization order, 19–20
 instance initializer, 18
 scope in, 31–34

bodies
 lambda expressions, 212,
 212
 methods, 171–172

boolean type
 default initialization, 31
 if statements, 70
 logical operators, 57–58
 size and range, 21
 ternary operators, 71–72
 while statements, 77
 wrapper classes, 134–135

braces ({})
 blocks, 18–19, 32–33
 if-then statements, 68
 lambda expressions, 212,
 212
 methods, 171
 try statements, 306–307

brackets ([]) for arrays, 7, 119–122

break statements
 in loops, 88–90, 88
 in switch, 74

byte type
 default initialization, 31
 promotion rules, 56
 size and range, 21
 switch statements, 72–73
 wrapper classes, 134–135

bytecode, 6

C

calling
 constructors, 243–244
 inherited class members, 244–245
 static variables and methods, 182–183

CamelCase names, 28–29

capacity vs. size, 113–114, 114

carets (^) for logical operators, 64, 64

case sensitivity of names, 27

case statements in switch, 73–76, 73

casting
 objects, 282–284
 operators, 60–61

catch blocks, 305–309, 305

chaining
 constructors, 201
 methods, 110–111
 StringBuilder, 112–113

char type
 default initialization, 31
 promotion rules, 56
 size and range, 21
 switch statements, 72–73
 wrapper classes, 134–135

charAt() method, 106, 114, 135

checked exceptions, 303, 305, 317

child classes in inheritance, 234

child packages, 10

.class extension, 6

class variables in default initialization, 30–31

ClassCastException class,
 315

classes
 abstract, 259–265
 concrete, 262–263
 description, 2
 element ordering, 34–35
 extending, 235–236, 235
 fields and methods, 2–4
 vs. files, 5
 immutable, 207–208
 inheritance. *See* inheritance
 interfaces. *See* interfaces
 packages, 9
 paths, 15
 subclasses, 319–321
 wrapper, 134–136

clear() method, 133

closures. *See* lambda expressions
 code

blocks, 18–19
 compiling, 6, 14–15
 exam formatting, 16

colons (:)
 dates and times, 150
 labels, 88
 paths, 15
 ternary operators, 71

commas (,)
 dates and times, 150
 exception lists, 171
 interface implementation, 267
 parameter lists, 171
 variable declarations, 26

- comments, 4
- comparators in arrays, 125
- compile-time constant values in switch statements, 73–76
- compiling code
 - extensions for, 6
 - and inheritance, 241–242
 - with packages, 14–15
- compound assignment operators, 62–63
- concatenating strings, 102–104
- concrete classes, 262–263
- conflicts in names, 12–13
- consistency of names, 28
- constant values in switch statements, 73–76
- constructors
 - calling, 243–244
 - chaining, 201
 - creating, 196
 - date and time, 141
 - default, 197–199
 - defining, 238
 - definition rules, 242–243
 - final fields, 202
 - initialization order, 202–204
 - objects, 17
 - overloading, 199–200
 - wrapper classes, 134–135
- contains() method
 - ArrayLists, 133
 - strings, 109–110
- continue statements, 90–91, 90
- control flow statements
 - break, 88–90, 88
 - continue, 90–91, 90
 - do-while, 78–80, 78
 - for, 80–83, 80, 85
 - for-each, 83–86, 83
 - if-then, 67–68, 67
 - if-then-else, 68–70, 69
 - labels, 87–89
 - nested loops, 87
 - switch, 72–76, 73
 - while, 76–77, 77, 79
- converting
 - ArrayLists to arrays, 136–137
 - number systems, 23
 - objects to wrapper classes, 135–136
- covariant return types, 248, 252
- curly braces ({})
 - blocks, 18–19, 32–33
 - if-then statements, 68

- lambda expressions, 212, 212
- methods, 171
- try statements, 306–307

D

- data types
 - in for statements, 82
 - promotion rules, 55–57
 - switch statements, 72–73
- Date class, 12–13
- dates
 - creating, 138–142
 - earlier versions, 141–142, 144, 150
 - formatting, 148–150
 - manipulating, 142–145
 - parsing, 151
 - periods, 145–147
- DateTimeFormatter class, 148–150
- dd format for dates and times, 150
- decimal numbering system, 22
- declare rules for exceptions, 303
- declaring
 - arrays, 121
 - methods, 166, 166
 - variables, 25
- decrement operators, 58
- default statements in switch, 72, 74
- defaults
 - access modifiers, 167, 175
 - constructors, 17, 197–199
 - interface methods, 274–278
 - packages, 13
 - variable initialization, 29
- defining
 - abstract classes, 260–262
 - interfaces, 267–269
- delete() method, 116
- deleteCharAt() method, 116
- descendants in inheritance, 234
- destroying objects
 - finalize(), 38–39
 - garbage collection, 36, 37–38
- diamond operator (<>), 130
- division
 - modulus operation, 54–55
 - precedence, 53–54
- do-while statements, 78–80, 78

dollar signs (\$) for names, 27, 29
 double quotes (")
 for spaces, 8
 strings, 102–103
 double type
 default initialization, 31
 size and range, 21
 wrapper classes, 134–135

E

else operators, 68–70, 69
 encapsulation
 description, 39
 immutable classes, 207–208
 overview, 205–206
 endsWith() method, 109
 enum type
 dates, 140
 support for, 72–73
 switch statements, 72
 equal signs (=)
 assignment operators, 60
 compound assignment operators, 62–63
 equality, 117–118
 equality operators, 65–66
 precedence, 53
 relational operators, 63
 unary operator, 57
 variable declarations, 25
 equality
 operators, 65–66
 overview, 117–119
 equals() method
 ArrayLists, 133–134
 arrays, 121
 strings, 109, 118–119
 equalsIgnoreCase() method, 109
 Error class, 302, 302, 305, 317–318
 exams, code formatting in, 16
 exception lists for methods, 171
 ExceptionInInitializerError class, 317
 exceptions
 checked, 317
 Error, 317–318
 finally blocks, 307–309, 307
 ignoring, 322
 methods, 318–322
 multiple, 311–313
 printing, 321, 322
 reasons, 300
 vs. return codes, 301–302
 roles, 300–301
 runtime, 314–316
 subclasses, 319–321
 throwing, 304–305
 try statements, 305–307, 305
 types, 302–303, 302, 309
 exclamation points (!)
 equality operators, 65–66
 logical operator, 57–58
 unary operator, 57
 “exclusive or” operator, 64, 64
 exit() method, 309
 extending
 abstract classes, 263–265
 classes, 235–236, 235
 interfaces, 269

F

false value in logical operators, 57
 fields
 in classes, 35, 41
 final, 202
 initialization order, 19–20
 overview, 2–4
 reading and writing, 18
 static, 181–182
 FileNotFoundException class, 317
 files vs. classes, 5
 final specifiers
 abstract classes, 261
 constants, 74
 constructor fields, 202
 interfaces, 267, 273–274
 methods, 168, 256
 finalize() method, 38
 finally blocks, 307–309, 307
 float type
 default initialization, 31
 promotion rules, 55
 size and range, 21
 wrapper classes, 134–135
 for statements, 80–83, 80, 85
 for-each statements, 83–86,
 83
 format() method, 148
 formatting dates and times, 148–150
 free store, 36

functional interfaces with lambda expressions, 214
 functional programming, 208–209
 functions. *See* methods

G

garbage collection, 36, 37–38
 gc() method, 36
 generics, 130
 getter methods in JavaBeans, 206
 getting variables, 18
 GMT (Greenwich Mean Time), 146
 greater than signs (>)
 diamond operator, 130
 lambda expressions, 211–212, 212
 relational operators, 63
 Greenwich Mean Time (GMT), 146

H

handle rules for exceptions, 303
 heap, 36
 hexadecimal number format, 22–23
 hh format for dates and times, 150
 hiding
 vs. overriding, 254–255
 static methods, 252–254
 variables, 257

I

identifiers, 27
 if-then statements, 67–68, 67
 if-then-else statements, 68–70, 69
 ignored returned values, 191
 IllegalArgumentException class, 315–316
 immutability of strings, 104–105
 immutable classes, creating, 207–208
 imports
 element ordering, 35
 packages, 9, 11–12
 static, 187–188
 “inclusive or” operator, 64, 64
 increment operators, 58
 indentation in if-then statements, 68

indexes
 arrays, 7, 120, 120
 strings, 105–107, 106
 indexOf() method, 106–107, 114
 infinite loops, 78
 infinite recursion, 317
 inheritance
 access modifiers, 237
 calling class members, 244–245
 compiler enhancements, 241–242
 constructors, 238, 242–244
 extending classes, 235–236, 235
 interfaces, 269–273, 277
 methods, 246
 object creation, 237–238, 238
 overview, 234, 235
 variables, 257
 initialization
 constructor order, 202–204
 default variables, 29
 object order, 19–20
 static, 186–187
 variables, 25–26
 initialization blocks in for statements, 80–82, 80
 initializer blocks, instances in, 18–19
 insert() method, 115
 instance members vs. static, 183–185
 instance variables
 default initialization, 30–31
 scope, 33
 instanceof operator, 63, 283
 instances
 creating, 17
 initialization order, 19
 initializer blocks, 18–19
 objects, 2
 instantiation process, 196
 int type
 default initialization, 31
 promotion rules, 55
 size and range, 21–22
 switch statements, 72–73
 wrapper classes, 134–135
 integrated development environment (IDE), 14
 interfaces
 abstract methods, 271–273
 with classes, 270
 default methods, 274–278
 defining, 267–269
 extending, 269
 implementing, 266, 266
 inheriting, 269–273

- lambda expressions, 214
- multiple inheritance, 277
- static methods, 278
- variables, 273–274

intValue() method, 135

IOException class, 317

isEmpty() method, 132–133

J

JAR files, 15

Java Development Kit (JDK), 6

- .java extension, 6
- java.lang package, 11

Java Runtime Environment (JRE), 6

Java Virtual Machine (JVM), 6

JavaBeans names, 205–206

javac command, 15

Javadoc comments

- description, 4
- key benefits, 39–40

JDK (Java Development Kit), 6

JRE (Java Runtime Environment), 6

JVM (Java Virtual Machine), 6

K

keywords, 3

L

L suffix for long, 22

labels, 87–89

lambda expressions

- example, 209–211
- predicates, 214–215
- syntax, 211–213, 212
- variable access, 213
- writing, 208–209

length() method, 106, 114

less than signs (<)

- diamond operator, 130
- relational operators, 63

lists. *See* ArrayLists

literal values

- primitive types, 22–23
- strings, 105, 117–118

- switch statements, 73–75

local variables, 29

LocalDate, 138–141, 146

LocalDateTime, 139–141, 146

LocalTime, 139–140

logical operators, 64

- complement, 57
- overview, 64–65
- precedence, 53

long type

- dates and times, 146
- default initialization, 31
- size and range, 21–22
- wrapper classes, 134–135

loops

- arrays, 123
- break statements, 88–90, 88
- continue statements, 90–91, 90
- do-while, 78–80, 78
- for, 80, 85
- for-each, 83–86, 83
- infinite, 78
- labels, 87–88
- nested, 87
- while, 76–77, 77, 79

M

main() method, 6

MAX_VALUE constant, 22

MEDIUM format for dates and times, 149

members in classes, 2

memory

- object destruction, 36, 37–38
- reference types, 24, 24

methods

- abstract, 259, 271–273
- access modifiers. *See* access modifiers
- ArrayLists, 130–134
- bodies, 171–172
- chaining, 110–111
- constructors. *See* constructors
- dates and times, 144
- declaring, 166, 166
- defining, 3
- designing, 166, 166
- exception lists, 171
- exceptions thrown by, 318–322
- final, 256
- hiding, 252–254
- interfaces, 271–279

- main(), 6
- names, 170
- optional specifiers, 168–169
- overloading, 191
- overriding, 246, 319–321
- overriding vs. hiding, 254–255
- overview, 2–4
- parameter lists, 171
- passing data, 188–191, 190
- polymorphism, 287–288
- redeclaring, 251–252
- return types, 169–170
- signatures, 3, 7, 166, 166
- static, 181–182
- StringBuilder class, 114–117
- strings, 105–109
- varargs, 172–173
- virtual, 284–285

minus signs (-)

- compound assignment operators, 62
- increment operators, 58–59
- lambda expressions, 211–212, 212
- negation operator, 57–58
- unary operator, 57

mm format for dates and times, 150

MMMM format for dates and times, 150

modulus operator (%)

- overview, 54–55
- precedence, 53–54

multidimensional arrays, 126–129, 127–128

multiple exceptions, 311–313

multiple inheritance

- description, 234–235, 235
- interfaces, 271–273, 277

multiple-line comments, 4–5

multiple variables, declaring, 26

multiplication, 53–54

mutability of StringBuilder, 112–113

N

names

- conflicts, 12–13
- constructors, 17
- identifiers, 27
- JavaBeans, 205–206
- methods, 170
- packages, 10, 12–13

native specifiers, 168

negation operator, 57

nested loops, 87

new keyword for constructors, 17

no-argument constructors, 198

NoClassDefFoundError class, 318

not equals operators, 65–66

now() method, 139

null values

- autoboxing, 136
- reference types, 24

NullPointerException class, 303, 316

NumberFormatException class, 316

numbering systems, 22–23

numeric promotion, 55–57

O

object-oriented languages, 39

objects

- casting, 282–284
- comparing, 66
- constructors, 17
- creating, 16, 237–238, 238
- description, 2
- destroying, 36–39, 37–38
- initialization order, 19–20
- instance initializer blocks, 18–19
- polymorphism, 281–282, 282
- primitive types, 20–23
- reading and writing fields, 18
- vs. references, 36, 37

octal number format, 22

ofLocalized methods, 149

operators

- arithmetic, 53–55
- assignment, 60, 62
- casting, 60–61
- equality, 65–66
- logical, 64–65, 64
- numeric promotion, 55–57
- order, 52
- overview, 52
- relational, 63
- ternary, 71
- unary, 57–59

optional specifiers, 168–169

“or” operator, 64, 64

order

- class elements, 34–35
- constructor initialization, 202–204
- field initialization, 19–20
- operators, 52

- overflow, 61
- overloading methods, 191
 - autoboxing, 193
 - constructors, 199–200
 - vs. overriding, 248
 - overview, 194–196
 - primitive types, 194
 - reference types, 193–194
 - varargs, 192–193
- overriding
 - vs. hiding, 254–255
 - methods, 246, 319–321
 - vs. overloading, 248
 - polymorphism, 287–288

P

- package private access modifiers, 167, 175
- packages
 - compiling code with, 14–15
 - creating, 13–14
 - element ordering, 35
 - names, 10, 12–13
 - overview, 9
 - redundant imports, 11–12
 - wildcards, 10–11
- parameters
 - main(), 7
 - methods, 3, 171
 - overloading methods, 192–193, 199
 - polymorphic, 285–286
- parent classes in inheritance, 234
- parentheses ()
 - lambda expressions, 212–213, 212
 - methods, 167
 - precedence, 54
- parse() method, 151
- parseInt() method, 135
- parsing dates and times, 151
- pass-by-reference languages, 190–191
- pass-by-value languages, 188–190, 190
- passing data in methods, 188–191, 190
- paths for classes, 15
- percent signs (%) for modulus, 54
- Period class, 146–147
- periods in dates and times, 145–147
- platform independence, 39–40
- plus methods for dates, 144
- plus signs (+)
 - compound assignment operators, 62
 - increment operators, 58–59
 - string concatenation, 103
 - unary operator, 57
- pointers, 24
- polymorphism
 - casting objects, 282–284
 - method overriding, 287–288
 - objects vs. references, 281–282, 282
 - overview, 279
 - parameters, 285–286
 - virtual methods, 284–285
- pools, strings, 105
- post-decrement operators, 58–59
- post-increment operators, 58–59
- pre-decrement operators, 58–59
- pre-increment operators, 58–59
- precedence of operators, 52
- predicates in lambda expressions, 214–215
- primitive types
 - arrays, 119–121, 119–120
 - casting, 60–61
 - overloading methods, 194
 - overview, 20–23
 - vs. reference, 25
- printing exceptions, 321, 322
- private access modifiers
 - abstract classes, 261
 - inheritance, 237
 - interfaces, 267–269
 - methods, 167
 - overview, 173, 174
- private methods, redeclaring, 251–252
- procedural languages, 39
- procedures. *See* methods
- promotion, numeric, 55–57
- properties in JavaBeans, 205
- protected access modifiers
 - interfaces, 267–269
 - methods, 167
 - overview, 176, 176
- public access modifiers
 - inheritance, 237
 - interfaces, 267–268, 273–274
 - methods, 167
 - overview, 7, 180–181

Q

question marks (?) for ternary operators, 71
 quotes (")
 for spaces, 8
 strings, 102–103

R

ranges of primitive types, 21
 readability
 if-then statements, 68
 literal values, 23
 names, 28
 polymorphic parameters, 286
 reading fields, 18
 recursion, infinite, 317
 recursive functions, 247
 redeclaring private methods, 251–252
 redundant package imports, 11–12
 references
 in arrays, 121, 122
 comparing, 66
 methods, 189–191, 190
 vs. objects, 36, 37
 overloading methods, 194
 overview, 24, 24
 polymorphism, 281–282, 282
 vs. primitive, 25
 relational operators
 overview, 63
 precedence, 53
 remainder operator (%)
 overview, 54–55
 precedence, 53–54
 remove() method, 131–132
 replace() method, 110
 reserved words, 27
 return codes vs. exceptions, 301–302
 return statement in methods, 169
 return types and values
 constructors, 17
 covariant, 248, 252
 description, 7
 ignored, 191
 immutable classes, 207–208

 methods, 169–170
 reverse() method, 116
 robustness, 40
 runtime exceptions, 314–316
 RuntimeException class, 302–303, 302, 305

S

scope
 in for statements, 81
 variables, 31–34
 searching arrays, 125–126
 security, 40
 semicolons (;)
 class paths, 15
 in for statements, 81
 lambda expressions, 212–213, 212
 variable declarations, 27
 set() method, 132
 setter methods in JavaBeans, 206
 setting variables, 18
 shift operators, 53
 short-circuit logical operators
 description, 64–65
 precedence, 53
 SHORT format for dates and times, 149
 short type
 default initialization, 31
 promotion rules, 56
 size and range, 21
 switch statements, 72–73
 wrapper classes, 134–135
 signatures in methods, 3, 7, 166, 166
 simplicity, 40
 single inheritance, 234, 235
 single-line comments, 4
 size
 vs. capacity, 113–114, 114
 primitive types, 21
 size() method, 132–133
 slashes (/)
 comments, 4–5
 division, 53
 precedence, 54
 sorting
 ArrayLists, 138

- arrays, 124–125
- spaces in arguments, 8
- square brackets ([]) for arrays, 7, 119–122
- StackOverflowError class, 318
- startsWith() method, 109
- statements
 - do-while, 78–80, 78
 - for, 80–83, 80, 85
 - for-each, 83–86, 83
 - if-then, 67–68, 67
 - if-then-else, 68–70, 69
 - overview, 66
 - switch, 72–76, 73
 - while, 76–77, 77, 79
- static keyword, 7
- static methods, hiding, 252–254
- static specifiers
 - calling variables and methods, 182–183
 - designing, 181–182
 - imports, 187–188
 - initialization, 186–187
 - vs. instance, 183–185
 - interfaces, 273–274, 278
 - methods, 168
 - variables, 185
- strictfp specifiers, 168
- String class, 102
- StringBuffer class vs, 117
- StringBuilder class, 111–112
 - creating, 113–114
 - methods, 114–117
 - mutability and chaining, 112–113
 - vs. StringBuffer class, 117
- strings
 - as arrays, 119
 - concatenating, 102–104
 - description, 3, 102
 - immutability, 104–105
 - methods, 105–109
 - pools, 105
 - StringBuilder class, 111–117
 - switch statements, 73
- subclasses with exceptions, 319–321
- substring() method, 107–108, 114
- subtraction
 - days, 145
 - precedence, 53–54
- super() keyword, 239–243, 246
- super keyword, 246–247

- switch statements, 72
 - compile-time constant values, 73–76
 - data types, 72–73, 73
- synchronized specifiers, 168
- System.gc() method, 36

T

- ternary operators
 - overview, 71
 - precedence, 53
- this() keyword, 238
- Throwable class, 302, 302
- throwing exceptions, 304–305
- times
 - creating, 138–142
 - formatting, 148–150
 - manipulating, 142–145
 - parsing, 151
 - periods, 145–147
- toEpochDay() method, 146
- toEpochTime() method, 146
- toLowerCase() method, 108
- toString() method, 105, 116–117
- toUpperCase() method, 108
- trim() method, 110
- true value in logical operators, 57
- truth tables, 64, 64
- try statements, 305–307, 305
- two-multidimensional arrays, 127

U

- unary operators
 - increment and decrement, 58–59
 - logical complement and negation, 57–58
 - precedence, 53
 - promotion rules, 56
 - working with, 57
- unchecked exceptions, 302
- underflow, 61
- underscores (_)
 - in literal values, 23
 - names, 27
- Unicode character set, 29
- update statements in for statements, 80–82, 80

V

valueOf() method, 135
variable argument (varargs) lists, 7–8
 ArrayLists, 137
 arrays, 126
 methods, 172–173
 overloading methods, 192–193
variables
 assignment operators, 60, 62–63
 declaring, 25
 default initialization, 29
 defining, 3
 description, 2
 in for statements, 81–83
 for-each statements, 83
 hiding, 257
 inheriting, 257
 interfaces, 273–274
 lambda expression access, 213
 reading and writing, 18
 scope, 31–34
 static, 185
vertical bars (|) for logical operators, 64, 64
virtual methods, 284–285
void type, 7
 description, 3
 methods, 169

W

while statements, 76–77, 77,
 79
wildcards
 JAR files, 15
 packages, 10–11
wrapper classes
 ArrayLists, 134–135
 converting to, 135–136
 switch statements, 73
“write once, run everywhere”,
 40
writing fields, 18

X

X for hexadecimal, 22

Z

ZonedDateTime, 139

