Index

Note to reader: **Bolded** page numbers refer to definitions and to main discussions of a topic. *Italicized* page numbers refer to illustrations.

### Symbols

- (decrement operator), 32, 32
& (AND) operator
  - bitwise operations, 51–53, 52
  - boolean operations, 54–56, 55
Q & A, 499, 504
! (boolean complement operator), 33–34
!= (inequality comparison operator), 50–51
$ (dollar sign), 5, 174
* (asterisk), import keyword and, 4–5
* (multiplication operator), 35–36
/ (division operator), 35–36
; (semicolon), 4–5
?: (conditional operators), 58–59, 63
[ ] (square brackets), 11
^ (XOR) operator
  - bitwise operations, 51–54, 52
  - boolean operations, 54–56, 55
_ (underscore), 5
{ } (curly brackets), 125, 207
+ (addition operator)
  - appending strings, 240
  - arithmetic operator, 37–39
  - unary plus operator, 32–33
++ (increment operator), 32, 32
< (less than) operator, 47–48
<< (left-shift) operator, 42–45
<= (less than or equal to) operator, 47–48
> (greater than) operator, 47–48
>= (greater than or equal to) operator, 47–48

### A

abstract classes, 289–290, 447–450
abstract modifiers
  - defined, 506
  - features of, 88
  - overview, 80–82, 81
Q & A, 90, 95
AbstractTableModel, 319–322, 322
access modifiers, 72–79
  - default, 75–76, 76
  - defined, 506
  - marking member classes with, 177
  - of overloaded methods, 166
  - overview, 72–73
  - private, 73–75
  - protected, 76–78
  - public, 73
Q & A, 92, 95
subclasses and method privacy, 78, 78–79
accessibility, method overriding, 168, 170
Accessible interface
  - building JTable, 316–317, 317
  - building JTree, 323
Action events
  - JTable, 319–322, 322
  - JTree, 323
Activatable object
  - exporting, 475, 475–479
  - Q & A, 483–484
active clients, 479–481, 480
add( ) method, 359
add(abstractButton) method, 309
add/remove( ) method pairs, 420–422, 426
addition operator (+)
  - appending strings, 240
  - arithmetic operator, 37–39
  - unary plus operator, 32–33
addXXXListener(XXXListener) method, 417
Adjustment events, JScrollPane, booleanValue() method

AND (&) operator
- bitwise operations, 51–53, 52
- boolean operations, 54–56, 55
- short-circuit logical operations, 56–58

anonymous classes
- construction/initialization of, 180–181
- defined, 506
- example, 182–183
- overview, 179–180

APIs, 269–270
append() method, 240
applets
- adding components to, 338
- Flow layout manager for, 343–345, 344–345
application partitioning, 441
args array, 14
arguments
- overriding vs. overloading, 166
- passing, 16–18, 18, 181

arithmetic error conditions, 40–41
- arithmetic-promotion conversions, 104, 104
- defined, 506
- error conditions, 40–41
- summary, 61
types
- addition and subtraction, 37–39
- modulo, 36–37
- multiplication and division, 35–36
arithmetic-promotion conversions
- defined, 506
- of operands, 46–47, 47
- overview, 103–104, 104
arrays
- Collections API storage, 243
data structures, 284
- defined, 506
- instanceof operator and, 49–50
- limitations, 241–242
- non-rectangular, 13
- object reference conversion, 109, 109–110

order of, 11–13, 12
Q & A, 24, 27, 487, 500
assertion
- defined, 506
- overview, 143–146
- Q & A, 155, 157
assignment conversion
- defined, 506
- literal values and, 102
- primitive, 99–102, 100–101
- Q & A, 118, 121, 494, 503
assignment operators, 59–60, 506
assignment, Developer's Exam. see Developer's Exam
asterisk (*), import keyword and, 4–5
automatic variables
- defined, 14, 506
- initialization, 15

B
bag, 242
Beans. see JavaBeans
bind() method, 400
binding, late, 169
bit size, 7–8, 26–27
bitwise inversion, 506
bitwise operators
- bitwise inversion operator, 33
- boolean operations, 54–56
- defined, 506
- overview, 51–54
Blocked state. see blocking
blocking
- Blocked state, 197–198, 198
- defined, 506
- overview of, 202–203, 203
- Q & A, 490, 501
boolean complement operator (!), 33–34
boolean literals, 9
boolean operations
- overview, 54–56, 55
- short-circuit logical operations, 56–58
booleanValue() method, 233
Border layout manager
  choosing, 286–287
  overview, 347–354, 348, 350–352, 354
boundary conditions, 161
brackets, curly (\{ \}), 125, 207
brackets, square ([ ]], 11
branch nodes, 323, 506
break statements, 124, 130

C
  callbacks, 286, 419, 507
  capacity, StringBuffer class, 238–239
Card layout manager, 286, 354–359
cascading style sheets, 279, 507
case labels, 132
case sensitivity, 275
cast operator, 34–35
casting
  defined, 507
  exam essentials, 115–116
  key terms, 116
  legal and illegal, 106–107
  object reference casting, 111–115, 112–113
  overview, 98
  primitive casting, 105–107, 106
Q & A, 119–121
catch blocks
  catching multiple exceptions, 136–137
  checking checked exceptions, 140–141
  defined, 507
  flow control in exceptions, 133–135, 135
Q & A, 151–153, 157
CellEditorEvent object, 316
char literals, 9–10
characters
  Developer's Exam coding tips, 275–276
  identifier, 5
  package name, 4
charValue() method, 233
check boxes
  adding to menu item, 316–318
  JCheckBox, 308–309, 309
checked exceptions
  checking, 140–141
  defined, 507
  method overriding and, 168
  overview, 138–140, 139
child nodes, 323, 507
class definitions, 4–5
class invariants, 145, 507
class locks, 213
class variables
  defined, 14, 507
  initialization, 15, 16
classes, 162–192
  constructors and subclassing
    overloading constructors, 172–173
    overview of, 170–171
  exam essentials, 183–184
  exam objectives, 159
  inner, 173–183
  constructing, 175–176
  defined inside methods, 177–178
  member classes, 176–177
  overview of, 173–175
  key terms, 184–185
  locks, 213
  main() method, 14
  Q & A, 185–191
  variables and initialization, 14–16
overloading methods
  invoking, 165–166
  method names, 164
  overview of, 163
  usefulness of, 164–165
overriding methods
  invoking, 169–170
  late binding, 169
  method, 166
  overview of, 163
  usefulness of, 166–168
relationships, 162–163
review Q & A, 185–191
CLASSPATH, 279, 473–475
client request structure, 387–394
client-server, 381–395
client request structure, 387–394, 388, 394
communication protocol, 384–397
defined, 507
Developer’s Exam, 274–275, 275
limitations, 394–395
overview, 467–468
Q & A, 412–413
server operation, 381–384
two-tiered database architecture, 440
client-server connections, 415–438
distributed notification, 434, 434
events
naming conventions, 417
notification, 417–419, 418–419
exam essentials, 435
Java Beans
conventions, 420–422, 422
MVC design pattern and, 423–425, 424–425
key terms, 436
listeners, 425–432, 429–430
overview, 416
remote notification, 432–433, 433
review Q & A, 437–438
summary, 435
code
Developer’s Exam, 269–270, 275–277
maintenance, 247–253
re-use of, 162
synchronized. see synchronized code
Collections API, 241–253
code maintenance, 247–253
data structures, 284, 457–458
defined, 242
equality and sorting, 244
hashCode( ) method, 245–246
implementation classes, 246–247
overview, 241–242
summary, 254
types, 242–444
columns
database design, 442–444
defined, 440, 507
Grid layout, 345–347, 346–347
GridBag layout, 359–365, 361–364
combo boxes, 313–314, 314
comma separators, 128
communication protocols
defining operations, 385
implementing, 385–387
overview, 384
Q & A, 412–413
compareTo( ) method, 244
comparison operators
defined, 507
equality, 50–51
ordinal, 47–50
compilation units
assertions and, 144
defined, 507
overview of, 4–5
components, 339
GUI requirements, 297
managing, see layout managers
Q & A, 334–335
types
container, 302–305
menu, 314–316
ordinary, 305–314
compound assignment operators, 59–60
concat( ) method, 240
concatenation
defined, 507
overloading + operator for, 38
overview, 240–241
concurrency, 380, 507
conditional operators (?:), 58–59, 508
conditions
defined, 508
for( ) loop, 127–128
connections. see client-server connections
constructors
building JTable, 316–318, 318
constructing String with, 236
String, 236
subclasing and, 170–173
container components
defined, 508
JFrame, 303–304, 303–304
JPanel, 304–305, 305
layout managers and, 338–341
overview, 302
synchronizing on, 457–458
content panes, JFrame
Border layout manager, 347–354, 348–354
JOptionPane, 331–332, 332
JSplitPane, 330–331, 331
using, 303–304
contexts, 419, 508
continue statement
defined, 124
Q & A, 149, 156
using in loops, 129–130
controllers
defined, 508
MVC design pattern, 423–424
conversion. see also casting
assignment conversion, 102
defined, 508
exam essentials, 115–116
explicit and implicit type changes, 98–99
key terms, 116
object reference conversion, 107–111
overview, 98
primitive conversion
arithmetic-promotion, 103–104, 104
assignment, 99–102, 100–101
method-call, 103
overview of, 99
review Q & A, 117–121
summary, 115
CORBA, 402
corner cases, 385
CounterThread, 195
coupling, 419, 508
curly brackets ({ }), 125, 207

D
data structures, 284, 458
data types
primitive, 6–9, 7–8
Q & A, 24, 27
signed integral, 7
database servers, 465–484
developing, 273–274, 274
exam essentials, 482
key terms, 482
record locking, 479–481, 480
requirements
exporting Activatable object, 475, 475–479
exporting with UnicastRemoteObject, 471–475
implementing RMI, 467–470
overview of, 466–467
review Q & A, 483–484
databases, 439–464
designing basic scheme
overview of, 441–442
using abstract classes, 447–450
using interfaces, 442–447
exam essentials, 460–461
implementation issues
design impediments, 454–455
exception handling, 451–454
supporting new features, 459–460
thread safety, 456–459
key terms, 461
review Q & A, 462–463
two-tiered, 440–441
DataProxy class, 433
Dead state, 197–198
deadlock, 215–217, 508
debugging
limitations of RMI, 401
print statements, 240
with toString( ) method, 229
declarations
anonymous inner classes, 180–181
array creation, 11–12
decrement operators (→), 32, 32
default access mode

features, 88
overriding, 78
summary, 79

default constructors, 170, 508
default packages, 279, 508
descriptors, 422
design patterns
defined, 508
follow-up exam, 287
MVC. see MVC (Model-View-Controller)
using standard, 276

Developer's Exam, 264–294
assignment
structure, 268–270
topics, 267–268
assignment example
client-server logic, 274–275
database/server development, 273–274
GUI development, 272–273
overview of, 270–272

coding tips
adhering to supplied naming, 275
stress readability, 275–276
using standard design patterns, 276
downloading assignment, 266–267
exam essentials, 291
follow-up exam, 267

follow-up exam preparation
abstract classes, 289–290
data structure, 284
design patterns, 287
exceptions, 285
layout managers, 286–287
listeners, 285–286
Thread class vs. Runnable interface, 284–285
using protected and default scope, 287–289

ggrading, 268
key terms, 291
readiness for, 264–265

review Q & A, 292–293
submission requirements
file structure, 279–280, 280
javadoc options, 278–279, 278–279
overview of, 276–278
using JAR tool, 282–283
writing README file, 281–282
dirty buffer, 460
distributed computing, 467–468, 508
distributed notification, 434, 434, 508
division operator (/), 35–36
do loops, 126
documentation
coding tips, 275–276
Developer's Exam, 278
doit() method, 217
dollar sign ($), 5, 174
double indirection, 17
DownCounter class, 196
downloading, Developer's Exam, 266–267

e else block, 131–132
empty for() loops, 128
encapsulation
defined, 509
overview, 160–161
preserving, 288
reuse and, 162
EnclosingClassName, 180
Enumeration interface, 459
equality comparison, 47, 50–51, 509
equals() method
constructing wrappers, 232
defining, 51
equality comparison operators, 50
hashCode() method, 245–246
Object class and, 228–229
String class, 235–236, 236
StringBuffer class, 240
ersors
arithmetic, 40–41
exception handling, 452
as exceptions, 138–139
overview, 509
evaluation order, 31
event adapters, 418
event handlers. see listeners
event notification
defined, 509
distributed, 434, 434
order of, 212–213
overview, 417–419
remote, 432–433, 433
event object, 417, 509
event source, 417, 509
event target. see listeners
events
defined, 509
JButtons, 307
JCheckBox, 308
JFrames, 304
JRadioButtons, 310
JScrollBars, 311
JTable, 316–317
JTree, 323
naming conventions, 417, 420–422
overview, 416–417
exams, Developer’s Exam. see
Developer’s Exam
exams, practice, 485–504
execute( ) method, 406
executeIfAvailable( ) method, 406
exceptions, 133–143
catching multiple, 136–137
categories, 138–139
defined, 509
follow-up exam, 285
handling, 451–454
method overriding and, 166
overview, 133
Q & A, 151, 154–155, 157, 462–463, 492, 502
throwing
checked exceptions, 138–141
overriding and, 141–143
throw statement, 137
throws statement, 137–138
try/catch/finally, 133–135, 135
execution, thread, 195–197
export( ) method, 398
exporting
Activatable object, 475–479
with UnicastRemoteObject, 471–475
expressions
defined, 509
for( ) loop, 127–128
for( ) loop and comma separator, 128
Q & A, 151, 156
extends clause, 163
extends keyword, 228
factory, 287
features
database, 459–460
defined, 72, 509
modifiers and, 72–73, 87–88
fields. see columns
file structures, 279–280, 280
fill feature
controlling component cell size, 371
GridBag layout, 368–369, 368–369
final modifier
defined, 509
features of, 88
Math class and, 229
overriding method and, 168
overview, 79–80
Q & A, 90, 95
finalize( ) method, 19
finally block
catching multiple exceptions, 136–137
flow of control in exceptions, 133–135
overview, 135
Q & A, 151–153, 157
firePropertyChange( ) method, 426–428
fireTableChange( ) method, 319, 321
flat data models
defined, 441, 510
designing basic scheme, 441–442
Q & A, 462–463
floating-point literals
arithmetic error conditions, 40
flow control – GridBag layout manager

defined, 8, 8, 510
overview, 10
flow control, 123–158
assertions, 143–146
eat exam essentials, 147–148
exceptions
catch, 136–137
finally, 135
overview of, 133
throwing, 137–143
try, 133–135, 135
key terms, 148
loop constructs, 124–130
break statement, 130
do loop, 126
empty for( ) loops, 128
for( ) loop, 126–128
for( ) loop and comma separator, 128
while( ) loop, 124–126
overview, 124
review Q & A, 149–157
selection statements
if/else construct, 131–132
switch( ) construct, 132–133
Flow layout manager, 286, 343–345, 344–345
follow-up exam
defined, 266
grading, 268
preparing for
abstract classes, 289–290
data structure, 284
design patterns, 287
exceptions, 285
layout managers, 286–287
listeners, 285–286
Thread class vs. Runnable interface, 284–285
using protected and default scope, 287–289
Q & A, 292–293
taking, 267
for( ) loops
comma separators and, 128
empty, 128
overview, 126–128
Q & A, 488, 500
frames
adding components to, 338–339
JFrames, 303–304, 303–304
JPanels, 304–305, 305
frameworks
Collections API, 242
defined, 510
free-floating initializer code, 85
friendly access mode. see default access mode

G

garbage collector
causing memory leaks, 20
defined, 510
overview, 19–20
Q & A, 26–27, 486, 489, 500–501
gc( ) method, 19
GeneralException, 137
get/set( ) method pair, 420–422, 426
getCommand( ) method, 393
getContentPane( ) method, 303–304
getData( ) method, 444
ggotLocation( ) method, 301
getPriority( ) method, 199
ggetSize( ) method, 301
ggotXXX( ) methods, 234
goto statement, 124
ggrading, Developer’s Exam
eease of use, 278
overview of, 268
subjecive process of, 266
ggraphical user interfaces. see GUIs
(graphical user interfaces)
ggreater than (>) operator, 47–48
ggreater than or equal to (>=) operator, 47–48
Grid layout manager
defined, 286
JPanels, 305
overview, 345–347, 346–347
GridBag layout manager, 359–374
component cell size, 369, 369–372
compoment position/stretch, 365–369, 367–368
defined, 286–287
layout design without, 359
overview, 359
rows and columns, 359–365, 361–362, 364
shorthand, 372–374, 374
GridColumnConstraints class, 359
gridheight
column component cell size, 369, 369–372
RELA TIVE and REMAINDER setting, 372–374, 374
gridwidth
column component cell size, 369, 369–372
RELA TIVE and REMAINDER setting, 372–374, 374
GUIs (graphical user interfaces), 295–336
Developer's Exam
client-server logic, 274–275, 275
overview of, 272–273, 273
exam essentials, 333
JMenus and actions, 327–329, 329
JTable
AbstractTableModel, 319–322, 322
overview of, 316–319, 317–318
JTree, 323, 323–327, 327
key terms, 333
overview, 296
panes, 329–332, 331–332
requirements, 297–300
identifying components, 297
isolating regions of behavior, 299–300
layout managers, 300
sketching, 298–299
review Q & A, 334–335
Swing components, 302–316
container components, 302–305
menu components, 314–316
ordinary components, 305–314
Swing methods, 300–302

H
"has a" relationships, 162–163
hash tables, 244, 510
hashCode( ) method, 245–246
HashMap class, 246, 284
HashSet, 246
Hashtable class, 246
height, grid, 369, 369–374, 374
hierarchy, 339, 510
HTTP servers, 381, 384–387
identifiers
defined, 510
overview, 6
Q & A, 24, 27
IETF (Internet Engineering Task Force), 381
if statements, 125, 129–130
if/else statements
boolean complement operator and,
33–34
conditional operator and, 59
overview, 131–132
implementation classes, 246–247
import statements, 4–5
increment operators (++), 32, 32
inequality comparison operator (!=), 50–51
inheritance, 162, 510
initialization
anonymous inner classes, 181
arrays and, 12, 12–13
Q & A, 494, 503
static, 85
variables and, 14–16, 15
inner class, 173–183
constructing, 175–176
defined, 511
defined inside methods, 177–183
member classes, 176–177
overview, 173–175
Q & A, 492–493, 502
innerMethod( ), 176
instance method, 83, 511
instanceOf operator, 48–50, 169
integral data types, 7–8, 40
integral literals, 10, 101
interfaces
abstract classes vs., 289–290
database design, 442–447
declaring method calls, 289
JTable, 317, 317
JTree, 323
listener, see listeners
maintenance benefit of, 450
object reference conversion, 109–110
Q & A, 462–463
intern( ) method, String class, 236–237
Internet Engineering Task Force (IETF), 381
interrupt latency, 46
interrupt( ) method, 197
"is a" relationships, 162–163
is/set( ) method pair, 426
isArray( ) method, 50
Item events
JCheckBox, 308
JRadioButton, 310
iteration expression, 127–128
iterator, 250, 511
Iterator interface, 459

J
JAR tool
command options, 282
file structure, 280
submitting Developer's Exam, 277
using, 282–283
JavaBeans
conventions, 420–422, 422
listeners, 425–432, 429–430
MVC design pattern and, 423–425, 424–425
java extension, 4
Java Programming Language Workshop, 265
java.lang package, 227–260
Collections API
code maintenance, 247–253
collection types, 242–244
equality and sorting, 244
hashCode( ) method, 245–246
implementation classes, 246–247
overview of, 241–242
summary, 254
exam essentials, 254
exam objectives, 227
key terms, 255
Math class, 229–230, 230
Object class, 228–229
review Q & A, 256–259
strings
concatenation, 240–241
String class, 235–238, 236, 238
StringBuffer class, 238–240, 239
wrapper classes, 231, 231–234
java.util package. see also Collections API
data structure choices, 284
java.util.Collection, 242
java.util.List, 243
java.util.Set, 243
javadoc, 277–279, 278–279
javax.swing package, 302
JButton component, 304, 306–308, 308
JCheckBox component, 308–309, 309
JCheckBoxMenuItem class, 316–317
 JComboBox component, 313–314, 314
JFileChooser, 329–332
JFrame component
adding components, 338
closing behavior, 318
inserting JMenuBar into, 316–318
overview, 303–304, 303–304
JIT (just-in-time) compiler, 289
JLabel component, 306, 306
JMenuBar component, 314–316, 316
JMenuComponent, 316–318
JMenus, 327–329, 329
JOptionPane, 331–332, 332, 430–432
JPanel component, 304–305, 305
JRadioButton component, 309–310, 310
JRadioButtonMenuItem class, 316–317
JScrollPane component, 310–311, 311
JSplitPane, 330–331, 331
JTable
AbstractTableModel, 319–322, 322
building, 316–319, 317–318
Q & A, 334–335
JTableHeader, 322
JTextArea component, 311–313
JTextComponent, 311–313
JTextField component, 311–313
JTree, 323, 323–327, 327
justification (alignment), 344–345, 511
just-in-time (JIT) compiler, 289

K
KeyStroke objects, 327–328
keywords, 5, 5–6

L
labels, 130, 492, 502
language fundamentals
argument passing, 16–18, 18
arrays, 11–13, 12
classes, 13–16, 15
exam essentials, 22–23
exam objectives, 3
garbage collection, 19–20, 20
key terms, 23
keywords and identifiers, 5, 5–6
literals, 9–11
primitive data types, 6–9
floating-point, 8
integral, 7–8
mathematical operations, 8–9
overview of, 7
review Q & A, 24–27
source files, 4–5
late binding, 169, 511
layout managers, 337–378
choosing for GUIs, 300
component size and position, 341–343, 342
exam essentials, 375
key terms, 376
other layout options, 374–375
overview, 286
policies. see layout policies
review Q & A, 377–378
theory, 338–341, 339
layout policies. see also GridBag layout manager
Card layout manager, 354–359
defined, 343
Flow layout manager, 343–345, 344–345
Grid layout manager, 345–347, 346–347
lazy clients, 479–481, 480
leaf node, 323, 511
left-shift (<<) operator, 42–45
less than (<) operator, 47–48
less than or equal to (<=) operator, 47–48
linked lists
Collections API, 246
defined, 243–244, 511
listeners
defined, 511
MVC design pattern, 424
overview, 423–432, 429–430
setting up relationships, 285–286
lists
Collections API, 242–243
defined, 511
ListSelectionEvent object, 316
literals
assignment conversion, 102
defined, 511
overview, 9–11
local variable, 511
lock( ) method, 479–481, 480
locks
class, 213
defined, 511
record, 479–481, 480
synchronization with object, 207, 207–208
loop constructs, 124–130
break statement, 129–130
continue statement, 129–130
do loop, 126
empty for( ) loops, 128
for( ) loop, 126–128
for( ) loop and comma separator, 128
while( ) loop, 124–126

Math class, 229–230
native modifier and, 85–86
overloading
invoking, 165–166
names, 164
overview of, 163
usefulness of, 164–165
overriding
invoking, 169–170
late binding, 169
method, 166
overview of, 163
usefulness of, 166–168
String class, 237–238
StringBuffer class, 239
subclasses and privacy, 78, 78–79
Swing, 300–302
synchronizing data access, 457
MIN_PRIORITY constant, 199
minus operator. see subtraction operator (−)
mixedUpMethod( ), 214
model transaction, 396
model-delegate, 423, 512
models, GUI, 273
Model-View-Controller. see MVC
(Model-View-Controller)
modifiers, 71–96
defined, 72, 512
exam essentials, 89
exam objectives, 71
features and, 87–88
key terms, 89
overview, 72
review Q & A, 90–95
static initializers, 85
summary, 88–89
types
abstract, 80–82, 81
access, 72–79
final, 79–80
native, 85–86
static, 82–85
synchronized, 87
transient, 86–87
volatile, 87
modulo operator  
defined, 512  
overview, 36–37  
Q & A, 486, 500
monitors, 205–218
  beyond the pure model, 213–215  
class lock, 213  
deadlock, 215–217  
defined, 205, 512  
defining, 197–198, 198  
monitor states, 204, 204
object lock and synchronization,  
  207–208  
overview, 205–206  
synchronizing code another way, 217–218
wait( ) and notify( ), 208–213, 209
multiplication operator (*), 35–36
multiset, 242
multithreaded programs, 194, 512
MVC (Model-View-Controller)  
  defined, 419, 512  
  Java Beans and, 423–425, 424–425
  Swing and, 296

N
Naming class, 400
naming conventions
  events, 417
  inner class, 174
  Java Bean, 420–422, 422
  overloading, 164
  overriding, 166–168
  tips for, 275
  using Card layout manager, 354
NaN (Not a Number)  
  floating-point calculations, 40–41
  primitive data types, 8
narrowing conversions
  casting requirements, 105–106, 106
  defined, 513
  overview, 102
  Q & A, 117, 121
native modifier  
  defined, 513

features, 88
overview, 85–86
negative numbers, shifting, 42–45
nested classes. see inner class
network protocols, 380–414
  client-server systems, 381–395
  client request structure, 387–394, 388, 394
  communication protocol, 384–387
  connecting clients to server, 384
  limitations, 394–395
  server operation, 381–384
exam essentials, 410
  key terms, 411
  overview, 380
review Q & A, 412–413
RMI, 395–402
  implementing, 397–401
  limitations, 401–402
  model transaction, 396, 396
  overview of, 395
sharing threads, 402–409
summary, 410
no-args (no-arguments)  
  constructors, 170
nodes, 323
non-rectangular arrays, 13
NORM_PRIORITY constant, 199
normalized exam questions, 264
Not a Number (NAN)  
  floating-point calculations, 40–41
  primitive data types, 8
notification, event. see event notification
notify( ) method
  lack of precision in, 213–214
  monitor states, 204
Object class, 228
  overview, 208–213, 209
synchronizing data access, 457
notifyAll( ) method
  client request structure, 390
  monitor states, 204
Object class, 228
  order of notification, 213–214
  synchronizing data access, 457
n-tier (three-tier) database model, 441, 513
NumberFormatException, 232, 234
Object class – overloading

Object class, 228–229
object comparisons, 50, 513
object locks, 207–208
object reference
  casting, 34–35, 111–115, 112–113
  conversion, 107–111
  defined, 513
  initialization values for, 15
  overview, 17
object serialization. see serialization
object-oriented. see also OO
  (object-oriented)
object-type comparisons, 47, 513
Observable interface, 286
Observer interface, 286
observer-observable, 419
on-demand clients, 479–481, 480, 483–484
OO (object-oriented). see also classes
  benefits
    encapsulation, 160–161
    inheritance, 162
    relationships, 162–163
  exam essentials, 183–184
  exam objectives, 159
  key terms, 184–185
  review Q & A, 185–191
operations, 385
operators, 29–70
arithmetic
  addition and subtraction, 40
  arithmetic error conditions, 40–41
  modulo, 36–37
  multiplication and division, 35–36
  assignment, 59–60
  bitwise
    boolean operations, 54–56, 55
    overview of, 51–54
comparison
  equality, 50–51
  ordinal, 47–50
  conditional, 58–59
  evaluation order, 31
  exam essentials, 63
  key terms, 64
  overloading, 165
  precedence order, 30
  review Q & A, 65–69
shift
  arithmetic promotion of operands, 46–47, 47
  fundamentals of, 41–42
  negative numbers, 42–45
  reduction of right operand, 45–46
  short-circuit logical, 56–58
summary, 61–63
unary
  bitwise inversion, 31–34
  boolean complement, 33–34
  cast, 34–35
  increment and decrement, 32, 32
  plus and minus, 32–33
OR operators
  bitwise operations, 51–54, 52
  boolean operations, 54–56, 55
  short-circuit logical operations, 56–58
ordinal comparisons, 47–50, 513
ordinary components, 305–314
  defined, 513
  JButton, 306–308, 308
  JCheckBox, 308–309, 309
  JComboBox, 313–314, 314
  JLabel, 306, 306
  JRadioButton, 309–310, 310
  JScrollPane, 310–311, 311
  JTextField and JTextArea, 311–313, 313
overflow conditions, 35
overloading
  addition operator, 38
  constructors, 172–173
  defined, 163, 513
  methods
    invoking, 165–166
    method names, 164
    overview of, 163
    Q & A, 185, 190
    usefulness of, 164–165
    Q & A, 488–489, 499, 500–501, 504
overriding

- defined, 163, 513
- exceptions, 141–143
- final modifier, 80
- methods
  - invoking, 169–170
  - late binding, 169
  - method, 166
- overview of, 163
- Q & A, 185, 190
- usefulness of, 166–168
- Q & A, 92, 95, 499, 504
- static modifier, 84
- subclasses, 78

method-call, 103
overview of, 99
primitive data types
overview, 6–9, 7–8
primitive conversion
  - arithmetic promotion, 103–104, 104
  - assignment, 99–102
  - method-call, 103
  - overview of, 99
  - wrapper classes, 231, 231–234
primitive values
  - casting, 34
  - creating reference to, 18
priority, 198–199, 514
privacy, 78, 78–79
private access modifier
  - defined, 514
  - features, 88
  - overview, 73–75
  - rules for overriding, 78
  - summary, 79
programming by contract, 140
properties
  - changes, 426–430
  - defined, 514
  - Java Bean naming conventions, 420–422
  - syntax supporting, 428
PropertyChangeEvent, 428–430
PropertyChangeListener, 426–428
PropertyChangeSupport class, 426–430
protected access
  - defined, 515
  - features, 88
  - overview, 76–78
  - rules for overriding, 78
  - summary, 79
protected identifiers, 287–288
public access modifier
  - defined, 515
  - features, 88
  - overview, 73
  - rules for overriding, 78
  - summary, 79
public classes, 4
publish-subscribe, 419
pushbuttons, 306–307, 307
right-shift (>>) operator

Q

Q & A
- casting, 117–121
- classes and objects, 185–191
- client-server connections, 437–438
- conversion, 117–121
- database servers, 483–484
- databases, 462–463
- Developer’s Exam, 292–293
- flow control, 149–157
- java.lang package, 256–259
- java.util package, 256–259
- language fundamentals, 24–27
- modifiers, 90–95
- network protocols, 412–413
- operators, 65–69
- practice exam, 485–504
- threads, 221–225
- user interface, 334–335

R

radio buttons
- adding to menu item, 316–318
- JRadioButtons, 309–310, 310
- raised exceptions, 515
- range. see bit size
- ray-tracing threads, 200–201
- RDMS (relational database management systems), 441
- read( ) call, 203
- README files
  - Developer’s Exam, 277–278
  - writing, 281–282
- Ready state
  - Blocked state transitions, 202–203, 203
  - defining, 197–198, 198
  - Monitor state transitions, 204, 204
  - monitor states, 204
  - Sleeping state to, 201–202, 202
  - yielding to, 199–201, 200
- rebind( ) method, 399–400
- records
  - defined, 440, 515
  - designing with interface, 443
  - locking, 479–481, 480
- reduction, 45–46
- redundancy brackets, 31
- reference
  - argument passing by, 16–18
  - comparisons, 50, 515
  - conversion, 107–111
  - defined, 515
  - registries, RMI and, 398–399
  - relational database, 440, 515
  - relational database management systems (RDMS), 441
- relationships, 162–163
- RELATIVE setting, 372–374, 374
- REMAINDER setting, 372–374, 374
- Remote interface, 468–470
- Remote Method Invocation. see RMI
  - (Remote Method Invocation)
  - remote notification, 432–433, 433
  - RemoteException, 397–398
  - RemoteObject, 397–398
  - removeXXXListener(KeyListener)
    - method, 417
- request-response services, 381, 467–468, 515
- requests
  - client request structure, 387–394, 388, 394
  - implementing RMI, 396–410
  - setting, 205
- reserved words, 5
- resume( ) method, 201
- retrieveMessage( ) method, 209–211
- return types
  - overloaded method, 165
  - overriding method, 166, 168
- re-use, code, 162
- right-shift (>>) operator
  - arithmetic promotion of operands, 46–47
  - reduction of, 45–46
  - shifting negative numbers, 42–45
RMI (Remote Method Invocation), 395–402
database server development, 273–274, 274, 467–470
implementing, 397–401
limitations, 401–402
model transaction, 396, 396
overview, 380, 395
Q & A, 412–413
remote notification with, 433, 433
rmid tool, 399
rmiregistry defined, 398
UnicastRemoteObject using, 475
roll back, 423, 428, 515
root nodes, 323, 515
rotate operation, 42
round-robin scheduling, 200–201, 515
rows defined, 515
layout
Flow layout manager, 343–345, 344–345
Grid layout manager, 345–347, 346–347
GridBag layout, 359–365, 361–364
relational database, 440
run( ) method
client request structure, 390–392
follow-up exam, 284–285
Q & A, 499, 503
sharing threads, 403–409
testing runtime behavior, 383
thread execution, 195–197
thread states at, 197
Runnable interface
follow-up exam, 284–285
Q & A, 412–413
sharing threads, 403–409
Running state
Blocked state transitions, 202–203, 203
defining, 197–198, 198
Monitor state transitions, 204, 204
Sleeping state transitions, 201–202, 202
yielding from, 199–201, 200
runtime exceptions
assertions, 143
defined, 516
exception handling, 452–453
overview, 138–139

S
scheduling. see thread schedulers
scopes, 287–288
scroll bars, 310–311, 311
Scrollable interface, 323
SecurityManager, 473
Seeking Lock state, 207, 207
selection statements, 131–133
semicolon (;), 4–5
serialization
FAQs, 402
goal of, 395
object, 394–395
Q & A, 412–413
servers
client-server. see client-server
connections. see client-server
connections
database. see database servers
Developer’s Exam, 273–274, 274
set, 516
setBackground( ) method, 301
setBounds( ) method, 342–343
setDaemon( ) method, 406
setEnabled( ) method, 301–302
setFont( ) method, 301
setLocation( ) method, 301
setPriority( ) method, 199
setSize( ) method, 301, 342
setVisible(false) method, 303
ShareThread class, 403–409
shift operators, 41–47
arithmetic promotion of operands,
46–47, 47
short-circuit logical operators — suspending

with bitwise inversion operator, 33
defined, 516
fundamentals, 41, 41–42
negative numbers, 42–45
reduction of right operand, 45–46
scenario, 43
short-circuit logical operators, 56–58
simple assignment operators, 59–60
single-threaded, 194, 516
size, component
controlling, 369, 369–372
overview, 341–343, 342
preferred, 343
skeletons
defined, 516
RMI transactions, 396, 473
sleep( ) method, 201–202, 202
sleeping threads
defined, 516
monitor states and, 204, 204
overview, 201–202, 202
Sleeping state, 197–198, 198
sorting collections
code maintenance, 251
overview, 244–245
sortStringArray( ) method, 250, 252
source files
overview, 4–5
Q & A, 24, 27
SpecificException, 137
square brackets ([ ]), 11
Stack class, 246
start( ) method, 195–197
statements. see also by type
defined, 516
using labels on, 130
states, thread
overview, 197–198
thread control and, 199
static initializers, 85, 516
static inner classes, 177
static methods, 176, 233–234
static modifier
defined, 516
features, 88
overview, 82–85
Q & A, 91–95
static variable. see class variables
stop( ) method, 197
storage
Collections API, 243–244, 254
Q & A, 258–259
storeMessage( ) method, 210–211
stretching, components, 365–369
String class
overview, 235–238, 236, 238
Q & A, 258–259
string context, 241, 516
string literals, 10–11, 516
StringBuffer class
overview, 238–240, 239
Q & A, 257–259
strings
concatenation, 240–241
constructing wrappers from, 232–233
converting operands to, 38–39
Q & A, 256, 259, 491–492, 502
String class, 235–238, 236, 238
StringBuffer class, 238–240, 239
stubs
database requirements, 473
defined, 396, 516
subclasses
constructors and, 170–173
defined, 163
defining classes, 143
method privacy and, 78, 78–79
thread execution and, 196
subtraction operator (-)
arithmetic, 40
unary, 32–33
super( ) reference
constructors, 171
overloading constructors, 172–173
Q & A, 187–188, 191
suspend( ) method, 201
suspending
defined, 517
Swing

overview, 201
Suspended state, 197–198, 198

Swing

common methods, 300–302
components, 302–316
counter components, 302–305
menu components, 314–316
ordinary components, 305–314
JMenus and actions, 327–329, 329
JTable
overview of, 316–319, 317–318
using AbstractTableModel,
319–322, 322
listeners, 425
Model-View-Controller design pattern,
423–425, 424–425
overview, 296
panes
JOptionPane, 331–332, 332
JSplitPane, 330–331, 331
switch( ) construct, 132–133
Sylvan Prometric centers, 267
synchronized code
another way, 217–218
client request structure, 390
defined, 517
object lock and, 207, 207–208
Q & A, 489, 501
thread safety, 456–459
synchronized modifier, 87, 88

T

TableColumnModeEvent objects, 316
TableModelEvent constructors, 316
tables, 440–441, 517
tags, JAR format, 283
targets, event, 418
telnet, 383–384
ternary operator (?:), 58–59, 517
this( ) reference
creating inner classes, 176
overloading constructors, 172–173
Q & A, 187–188, 191
thread pool, 403, 517
thread schedulers
defined, 195, 517
determining priority, 198–199
implementing, 204–205
threads, 193–226
controlling
blocking, 202–203, 203
monitor states, 204
scheduling implementations, 204–205
sleeping, 201–202, 202
suspending, 201
yielding, 199–201, 200
exam essentials, 219
follow-up exam, 284–285
fundamentals
execution, 195–197
priority, 198–199
states, 197–198
when execution ends, 197
key terms, 220
monitors
beyond pure model, 213–215
class lock, 213
deadlock, 215–217
object lock and synchronization, 207,
207–208
overview of, 205–206
synchronizing code, 217–218
wait( ) and notify( ) methods,
208–213, 209
Q & A, 498–499, 503
review Q & A, 221–225
sharing, 402–409
summary, 218–219
threadsafẽ
defined, 242, 517
overview, 456
Q & A, 462–463
three-tier (n-tier) database model,
441, 513
throw statement
overview, 137
Q & A, 492, 502
Throwable class, 452
throwing exceptions – vetoable

throwing exceptions, 137–143
checked exceptions
checking, 140–141
overview of, 138–140, 139
defined, 517
overriding and, 141–143
throw statement, 137
throws statement, 137–138
throws statement
checking checked exceptions, 140–141
overview, 137–138
time-sliced scheduling
overview, 200–201
Q & A, 489, 501
toString( ) method
converting operands to strings, 38–39
exception handling with, 453
string concatenation, 229, 240–242
wrapper classes providing, 234
traceRays( ) method, 200–201
transient modifier
defined, 517
features, 88
overview, 86–87
Q & A, 91–92, 95
TreeMap class, 246, 251–252
trees
building JTree, 323–327
Collections API storage, 244
defined, 517
TreeSet class, 246
try block
catching multiple exceptions, 136–137
checking checked exceptions, 140–141
defined, 517
flow of control in exceptions, 133–135, 135
Q & A, 151–153, 157
try/catch/finally construction,
133–135, 135
two-tiered databases, 440–441, 517

U

unary operators
arithmetic-promotion conversions,
104, 104
bitwise inversion operator, 33
boolean complement operator,
33–34
cast, 34–35
defined, 517
increment and decrement operators,
32, 32
overview, 31–32
plus and minus operators, 32–33
summary, 61
underflow condition, 36
underscore (_), 5
UnicastRemoteObject, 471–475,
483–484
UnknownException, 137
unlock( ) method, 479–481, 480
uploading, Developer’s Exam, 277
user interfaces. see GUIs (graphical user interfaces)

V

values
argument passing, 16–18
assignment operators, 60
Q & A, 486, 500
vectors
adding object to, 233
choosing for data structures, 284
Collections API, 246
vetoable, 426, 518
VetoableChangeSupport class, 426–428
volatile modifier
    defined, 518
features, 88
overview, 87

W
wait( ) method
    client request structure, 390
monitor states, 204
Object class, 228
overview, 208–213, 209
synchronizing data access, 457
waiting state, 198, 204, 518
Web site information
    Developer's Exam for Java 2, 267
event notification, 419
Java 3 registry service, 399
Java Programming Language Workshop, 265
RMI implementation, 402
syntax supporting properties, 428
uploading assignment, 277
weight
    defined, 518
GridBag layout manager, 360, 363–365
well-known port, 381, 518
while( ) loop
    comparing to do loop, 125
    comparing to for( ) loop, 127–128
    overview, 124–126

whitespace
    Q & A, 495–496, 503
    in source files, 5
widening conversions
    casting and, 106
    defined, 518
    overview, 100–101
    Q & A, 117, 121
width
    defined, 518
    grid, 369, 369–372, 374
    Q & A, 489, 501
Window events, 304
windowClosing( ) method, 304
wrapper class, 231, 231–234, 518

X
XOR (eXclusive-OR) operators (^)
    bitwise operations, 51–54, 52
    boolean operations, 54–56, 55
XXXValue() methods, 234

Y
yield( ) method, 199–201, 200
yielding
    defined, 518
    overview, 199–201, 200