Symbols

&& (ampersands), and operator, 209–210, 212, 213
* (asterisk), multiplication operator, 36, 137
\ (backslash)
  escape sequence, 156, 523
  in file names, 523
{ } (braces)
  enclosing blocks of code, 201–202
  lining up code, 181
< > (brackets), diamond syntax, 352
$ (dollar sign), in variable names, 37
. (dot), name syntax, 403
= (equal sign), assignment operator, 38–39
== (equal signs), relational operator
  comparing object references, 187
  comparing strings, 186, 189–190
  equality testing, 184–185
  versus equals method, 186
  syntax, 184
  testing for null, 187–189
! (exclamation point), not operator, 211
!= (exclamation point, equal), relational operator, 184
> (greater than), relational operator, 184
>= (greater than or equal), relational operator, 184
-> (hyphen, greater than), arrow, 485, 855
< (less than), relational operator, 184
<= (less than or equal), relational operator, 184
- (minus sign), subtraction operator, 36
-- (minus signs), decrement, 138
( ) (parentheses)
  in arithmetic operations, 137, 142
  in object construction, 47
  balancing, 142
% (percent sign), modulus
  avoiding negative remainders, 143
  description, 138–139

full code example, 140
+ (plus sign)
  addition operator, 36
  concatenation operator, 155
++ (plus signs), increment, 138
" (quotation marks), string delimiters, 12
/ (slash), division operator, 36, 137
// (slashes), comment delimiter, 38
/* ... */ (slash asterisk ...), comment delimiters, 38
/** (slash, asterisks), comment indicator, 88
; (semicolon)
  ending if statements, 182
  ending method statements, 11
  omitting, 13
_ (underscore), in variable names, 37
|| (vertical bars), or operator, 209–210, 212, 213

A
abs method, java.lang.Math class, 140, A-18
abstract classes, 443–444
abstract data types, 741
abstract methods, 443–444
access specifiers, 81
accessor methods
  definition, 379–380
  description, 48–49
  full code example, 49
acos method, java.lang.Math class, A-19–20
actionPerformed method, java.awt.
  event.ActionEvent interface
  creating buttons, 496–498, 499
  description, A-14
  reading text fields, 889–890
actor classes, 376
add method
  java.awt.Container class, A-13
  java.math.BigDecimal class, 136, A-22
  java.math.BigInteger class, 136, A-22
  java.util.ArrayList<E> class, 344, A-26
  java.util.Collection<E> interface, A-26
  java.util.Iterator<E> interface, 688
  java.util.Queue<E> interface, 699
  javax.swing.ButtonGroup class, A-33
  javax.swing.JMenu class, A-34
  javax.swing.JMenuBar class, A-34
  addChangeListener method, javax.
    swing.AbstractButton class, 492–498, A-33
  addChangeListener method, javax.
    swing.JSlider class, 914–916, A-35
  addFirst method, java.util.
     LinkedList<E> class, 723–724, A-28
  addItem method, javax.swing.
     JComboBox class, A-33
  addMouseListener method, java.awt.
     Component class, 503, A-12
  addLast method, java.util.
     LinkedList<E> class, 730–731, A-28
  addMouseListener method, java.awt.
     Component class, 503, A-12
  addNode method, Node class, 778
Address class, 585–586
aggregation, 570–571, 574–575
algebraic expressions
  evaluating with stacks, 703–706
  full code example, 706
  precedence, 704
algorithms. See also loops, common
algorithms; streams, common
algorithms.
  characteristics of, 16
  combining and adapting, 327–332
  definition, 16
  developing, 332–335
  evolving into programs, 18–19
  example, 17

algorithms (continued)
  executable, 16
  full code examples, 329, 335
  pseudocode for, 19
  reusing, 327–332
  terminating, 16
  for tiling a floor, 21–22
  unambiguous, 16

allMatch method, java.util.stream.
  Stream<T> interface, 862, A-32

Altair 8800, 407
American National Standards
  Institute (ANSI), 686
ampersands (&&, and operator, 209–210, 212, 213
Andreesen, Marc, 456
anonymous objects, 488–489
anonymous classes, 488–489
ANSI (American National Standards
  Institute), 686
anyMatch method, java.util.stream.
  Stream<T> interface, 862, A-32
API (application programming
  interface), 50–53
append method, javax.swing.JTextArea
  class, 891–893, A-35
Apple II, 407
applets, definition, 6
application development, examples.
  See “Hello, World” program; printing an invoice (sample program).
application programming interface
  (API), 50–53
approximate equality, 185
approximate solutions, finding, 281–282
AreaMeasurer.java class, 484
arguments
  definition, 12
  method, 42–43
  object construction, 47
Ariane rocket incident, 554
arithmetic operations. See also data
types; specific operations.
  combining with assignment, 143
  computation overflow, 131
  hand tracing, 152–154
  integer division, 138–139, 142
  mathematical methods, 140
  modulus, 138–139
  powers, 139–140
  remainders, 138–139
  roots, 139–140
  rounding, 131–132
  roundoff errors, 132
  unintended integer division, 142
arithmetic operators. See also specific
  operators.
  * (asterisk), multiplication, 36, 137
  - (minus sign), subtraction, 36
  -- (minus signs), decrement, 138
  % (percent sign), modulus, 138–139
  + (plus sign), addition, 36
  ++ (plus signs), increment, 138
  / (slash), division, 36, 137
  definition, 137
  expressions, 137
  parentheses, 137, 142
  precedence, 137, A-5
ARPANET, 456
array lists. See also java.util.
  ArrayList<E> class.
  versus arrays, 350–351
  auto-boxing, 347
  constructor, 344
  converting from array algorithms, 348
  copying, 346
  declaring, 344–345
  definition, 343
  diamond syntax, 352
  enhanced for loop, 345–346
  full code example, 349
  generic class, 344
  inserting elements, 348
  iterating through, 345–346
  overview, 344–345
  removing matches, 348–349
  size, determining, 352
  syntax, 343
  type parameter, 344
  wrapper classes, 347, 347t
array lists, implementing
  adding/removing elements, 738–739
  efficiency, 740t
  full code example, 740
  getting/setting elements, 737–738
  growing the internal array, 739–741
array store exception, 833–834
ArrayList constructor, A-26
arrays. See also enumeration types.
  versus array lists, 350–351
  bounds errors, 310, 314
  companion variable, 312
  converting to streams, 847–848
  creating from streams, 850
  current size, 312
  declaring, 308–311
  definition, 308
  filling, 311, 868–869
  fixed length, 311
  full code example, 313
  How To example, 330–332
  implementing stacks as, 743
  initialization, 308–310
  iterating through, 317–318
  length, determining, 312–313
  with methods, 312
  multidimensional, 343
  numbering elements of, 310
  of objects, 314–315
  overview, 308–310
  parallel, 314–315
  partially filled, 312–313
  printing, 340–341
  references, 311
  resizing, 311, 312–313
  sequences of related values, 314
  size, determining, 352
  sorting, 327
  syntax, 309
  underestimating data size, 327
  unfilled, 314
  uninitialized, 314
  variable row lengths, 341–342
  world population table (Worked
    Example), 341
arrays, common algorithms
  averaging elements, 319
  binary search, 320
  converting array algorithms to
    array lists, 348
  copying arrays, 323–324
  element position, finding, 320
  element separators, 319–320
  filling, 318
  growing arrays, 323–324
  inserting elements, 321
  linear search, 320
  maximum value, finding, 319
  minimum value, finding, 319
  reading input, 324–326
  removing elements, 320–321
rolling dice (Worked Example), 332
searching for values, 320
sequential search, 320
summing elements, 319
swapping elements, 322
arrays, two-dimensional
accessing elements, 337–338
accessing rows and columns, 338–339
declaring, 336–337
definition, 336
locating neighboring elements, 338
syntax, 337
ArrayUtil.java class, 638–639
artificial intelligence, 217
ASCII encoding, 524–525
assertions, 549
assignment = (equal sign), assignment operator, 38–39
changing variable values, 38–39
combining with arithmetic operations, 143
versus declaration, 40–41
full code example, 39
syntax, 39
association, 574–575
associativity, Java operators, A-5–6t
asterisk (*), multiplication operator, 36, 137
asymmetric bounds, 256
atan method, java.lang.Math class, A-19–20
atan2 method, java.lang.Math class, A-19–20
ATM simulation (Worked Example), 587
attributes, UML, 573
Augusta, Ada, 658
auto-boxing, 347
AutoCloseable interface, 545–546, A-17
average method, primitive streams, 865
averages
array elements, 319
computing with loops, 268
stream algorithms, 869
streams, 866–868
averagingDouble method, java.util.
stream.Collectors class, 867, A-31
averagingInt method, java.util.
stream.Collectors class, 867, A-31
averagingLong method, java.util.
stream.Collectors class, 867, A-31
await method, java.util.concurrent.
locks.Condition interface, A-30
Babbage, Charles, 658
baby name analysis (Worked Example), 538
backing up files, 10
backslash (\), escape sequence, 156, 523
in file names, 523
backtracking
definition, 620
Eight Queens problem, 621–626
Four Queens problem, 622–623
full code example, 708
with stacks, 706–708
BadDataException.java class, 553–554
balanced trees, 772–773
balancing parentheses, 701
BankAccount.java class
full code example, 89
sample code, 94–95
BankAccountTester.java class, 101
batch files, 354
Berners-Lee, Tim, 456
BigDecimal type, 136, A-22
BigInteger type, 136, A-22
big-Oh notation
analyzing selection sort performance, 642–644
common growth rates, 645t
definition, 644
measuring selection sort performance, 642–644
binary search, 320
binary search trees
binary search property, 775–776
code sample, 780–783
definition, 775
efficiency, 780, 780t
generic classes (Worked Example), 839
inserting nodes, 776–778
insertion points, 778
removing nodes, 778–780
self-organizing structure, 778
binary searching, 655–658
binary trees
balanced trees, 772–773
decision trees, 770
definition, 770
expression trees, 771
full code example, 774
Huffman trees, 771, (Worked Example) 774
implementing, 773–774
binarySearch method
java.util.Arrays class, 664–665, A-26
java.util.Collections class, 664–665, A-27
BinarySearcher.java class, 656–657
BinarySearchTree.java class, 780–783
black boxes, 83
black-box testing, 206
blocks of code
definition, 201–202
enclosing with braces, 201–202
BlueJ environment, 54–55
Booch, Grady, 378
boolean data type, 130t
Boolean constructor, A-17
Boolean operators
& (ampersands), and operator, 209–210, 212, 213
! (exclamation point), not operator, 211
| (vertical bars), or operator, 209–210, 212, 213
De Morgan’s Law, 213
flags, 209
full code example, 210
inverting conditions, 211
precedence, 210
short circuiting evaluation of, 213
Boolean variables, in loops, 261
booleanValue method, java.lang.Boolean class, A-17
border layout, 884
BorderLayout constructor, 884–885, A-12
boundaries, radio buttons, 894–895
boundary test cases, 206–207
bounds errors, arrays, 310, 314
bounds for loops, choosing, 256
braces ({ }), coding guidelines, A-44
enclosing blocks of code, 201–202
lining up code, 181
brackets (< >), diamond syntax, 352
breadth-first search, 787–789
break statements, 263–264
breakpoints, 283–285, 598–599
buckets, 748
buffer overrun attacks, 316
bugs, historical first, 287. See also debuggers; debugging.
button-press events, 496–498
buttons (user interface), 496–498
ButtonViewer.java class, 492–493
byte data type, 130
calculator program full code example, 886
CalculationExample.java class, 702–703
calendar programs, calculating elapsed days (Worked Example), 55
call stack, 599
callbacks, 482–486
call-by-reference, 382–386
call-by-value, 382–386
calling methods, 32–33
camel case, 37
CamelComponent.java class, 112
Cars.java class, 112–113
cars
autonomous vehicles, 217
drawing, 110–113
CarViewer.java class, 113
case, transforming, 852
case sensitivity
constants, 133
definition, 9
errors caused by, 15
misspelling words, 15
variable names, 37
CashRegister.java class
full code example, 104
sample code, 134–135
CashRegisterTester.java class, 135–136
cast operator
converting floating-point numbers to integer, 140–141
full code example, 140
syntax, 141
casting from interfaces to classes, 476–477
catch clause, 542–543, 548
catching exceptions, 542–543
ceil method, java.lang.Math class, 140t, A-19–20
central processing unit (CPU), 3
char data type, 130
character encodings, 524–525
character patterns. See regular expressions
characters. See also strings.
classifying, 526–527
definition, 154, 156
international alphabets, 161
reading, 526
reading from a string, 528
versus strings, 156–157
Unicode, 156–157, 161, A-1
charAt method, java.lang.String class, 156
check boxes, 895–896
checked exceptions, 543–545
child nodes, 767
ChoiceQuestion.java class, 435–436
circles, drawing, 64–65
circular arrays, implementing queues as, 744–746
CircularArrayQueue.java class, 745–746
class files, 9
class relationships
aggregation, 570–571, 574–575
association, 574–575
composition, 574–575
coupling, 569–570
dependency, 569–570
full code example, 570
“has a”. See aggregation.
inheritance, 571–572
“is a”. See inheritance.
“knows about”. See dependencies, classes.
multiplicities, 574
recording, sample program, 578–579
classes. See also subclasses;
superclasses; specific classes.
abstract, 443–444
actors, 376
anonymous, 488–489
candidates for, 567
cohesion, 377–378, 381–382
collaborators, 568
concrete, 443–444
consistency, 381–382
CRC (class-responsibility-collaborator) method, 567–569, 572–573
declaring, 11
definition, 11, 33
dependencies, 378–379, 381–382
discovering, 376–377, 566–567
for drawing shapes, 110–114
extending. See inheritance final, 444
identifying, 566–567
immutable, 379–380
importing from packages, 52
inner, 487–488
interfaces, converting from, 476
naming conventions, 37
nouns as, 566
public. See public interfaces, classes
public interfaces, 41–42
testing, 54–55
utility, 376
classes, common patterns
collecting values, 387–388
counters, 387
full code example, 390
managing object properties, 388
modeling moving objects, 389–391
modeling objects with distinct states, 388–389
summing totals, 386–387
classes, implementing
constructors, 92–93
How To, 96–99
instance variables, 91
for a menu (Worked Example), 99
methods, 93–95
class-responsibility-collaborator (CRC) method. See CRC (class-responsibility-collaborator) method.
click method, 82
ClickListener.java class, 492
clean method, java.lang.object class, 479–481, A-20
Cloneable interface, 479–481

close method
java.io.InputStream class, A-16
java.io.OutputStream class, A-16
java.io.ObjectInputStream class, A-16
java.io.ObjectOutputStream class, A-16
java.io.PrintStream class, A-16
java.io.PrintWriter class, 521–522, A-16
java.lang.AutoCloseable interface, A-17
java.net.ServerSocket class, A-22
java.net.Socket class, A-22
java.sql.Connection interface, A-24
java.sql.ResultSet interface, A-24
java.sql.Statement interface, A-25
java.util.Scanner class, A-30
code coverage, 206
coding guidelines
{} (braces), A-44
classes, A-40
constants, A-41–42
control flow, A-42–43
for statement, A-42
indentation, A-44
methods, A-41
naming conventions, A-44
overview, A-39
source files, A-40
unstable layout, A-45
variables, A-41–42
white space, A-44
codePoints method, java.lang.String class, 849, 864, 869, A-21
cohesion, classes and methods, 377–378, 381–382
collaborators, 568
collect method, java.util.stream
Stream<T> interface, 850, A-32
collecting values, common class patterns, 387–388
collections
creating from streams, 850
definition, 678
versus streams, 847
collections framework
choosing (How To), 694–695
definition, 678
full code example, 680
lists, 678–679
maps, 679–680, 692–694
priority queues, 679, 699–700
queues, 679, 699–700
sets, 679, 687–691
stacks, 679, 698
collectors, streams, 866
Collectors.joining method, 850
collisions, 696–697, 747
Color constructor, A-12
ColorFrame.java class, 915–916
colors, 66, 667
ColorViewer.java class, 915
combining strings, 850–851
combo boxes
code samples, 897–900
definition, 896
overview, 896–897
command line arguments, 533–535
comments
// (slashes), delimeter, 38
/*...*/ (slash asterisk...), delimiters, 38
converting to documentation. See javadoc
definition, 38
documentation, 88–90
public interfaces to classes, 87–90
on variables, 38
commit method, java.sql.Connection interface, A-24
companion variable, 312
Comparable interface, 477–479
Comparator.comparing function, 859
comparators with lambda expressions, 667
compare method
java.lang.Double class, A-17
java.lang.Integer class, A-17
java.util.Comparator<T> interface, 666–667, A-27
compareTo method
comparing objects, 477–478
comparing strings, 186
java.lang.Comparable<T> interface, 665, A-18
java.lang.String class, A-20
returned values, 666
comparing method, java.util.Comparator<T> interface, 667, A-27
comparisons
adjacent values, with loops, 271–272
adjacent values, with stream algorithms, 870
approximate equality, 185
floating-point numbers, 185
full code example, 187–189, 665
lexicographic ordering, 186
object contents, 187, 477–478. See also compareTo method; equals method.
object references, 187
objects, 665–667
rectangles, 187
relational operators, 184–185, 212
strings, 186, 189–190
syntax, 184
testing for null, 187–189
compilation process, 9
compilers, 6
compile-time errors, 14
composition, 574–575
computation, limits of, 612–613
computation overflow, 131
computer programs, definition, 2. See also Java programs; programming; software.
computer viruses, 316
countries
 anatomy of, 3–5
common uses for, 5
description, 2
different uses of. See history of computers.
PCs (personal computers), schematic design of, 4
concatenating strings, 155
concrete classes, 443–444
conditional operators, 182–183
consecutive integers, primitive streams, 863–864
consistency, classes and methods, 381–382
console input, 155
constants
case sensitivity, 133
declaring, 134
definition, 132
in interfaces, 473
naming conventions, 133
syntax, 134
constructing objects. See object construction.
constructor references, lambda expressions, 857–858
constructors
calling one from another, 110
declaring as void, 90
full code example, 48
implementing classes, 92–93
invoking like methods, 48
versus methods, 85
naming, 86
no-argument, 86
specifying, 85–86
subclasses, 438
superclasses, 438
containers
definition, 884
as event listeners, 499
contains method, java.util.
Collection<E> interface, 688,
A-27
continue statement, 263–264
converting. See also cast operator.
array list algorithms from array
algorithms, 348
classes to interface variables, 475
comments to documentation,
88–90
floating-point numbers to integer,
140–141
interfaces to classes, 475
numbers from strings, 528–529.
See also parseInt method.
stems to arrays or collections,
850–851
strings from objects. See toString
method.
strings to numbers, 160, 528–529.
See also parseInt method.
copy method, java.nio.files.Files.
class, A-23
copying. See also java.lang.Cloneable
interface.
array lists, 346
arrays, 323–324
files. See backing up files.
objects, 477–481. See also clone
method.
copyOf method, java.util.Arrays.
class, 323–324, A-26
cos method, java.lang.Math class,
140t, A-19–20
count method, java.util.stream.
Stream<T> interface
counting words, 846
description, A-32
as terminal operation, 862
count-controlled loops, 250
Counter.java class, 83–84
full code example, 83
methods, 82
counters
common class patterns, 387
in loops, 268–269
counting
items in a list, 866–868
loop iterations, 256
matches, with stream algorithms,
869
words, 846
counting method, java.util.stream.
Collectors class, A-31
CountryValue.java class, 537–538
coupling, 569–570
CPU (central processing unit)
definition, 3
simulating. See virtual machines.
CRC (class-responsibility-
collaborator) method
discovering classes, 576–578
in program design (How To),
572–573
createDirectory method, java.nio.
files.Files class, A-23
createElement method, org.w3c.dom.
Document interface, A-37
createFile method, java.nio.files.
Files class, A-23
createLSerializer method, org.
w3c.dom.ls.DOMImplementationLS
interface, A-37
createStatement method, java.sql.
Connection interface, A-24
createTextNode method, org.w3c.dom.
Document interface, A-37
credit card processing (Worked
Example), 275
currentTimeMillis method, java.lang.
System class, 640–641, A-21
Curry, Haskell, 856
customer queues, simulating
(Worked Example), 708
CYC project, 217
D
dangling else problem, 201
DARPA, 456
Urban Challenge, 217
data types. See also arithmetic
operations; specific types.
abstract, 741
boolean, 130t
byte, 130t
char, 130t
double, 130t, 131–132
float, 130t
int, 36, 130t
long, 130t, 131
number literals, 131
numbers with fractions. See
double data type; floating-
point numbers.
numbers without fractions. See
int type.
short, 130t
testing for, 453–455. See also
instanceof operator.
DataAnalyzer.java class, 550–551
databases, privacy issues, 586
Data.java class, 470, 484
DataSetReader.java class, 556–557
Date constructor, A-27
De Morgan, Augustus, 213
De Morgan’s Law, 213
debuggers
breakpoints, 283–285
definition, 282
inspecting variables, 283–285
overview, 282–285
single stepping, 283–285
single-step command, 284
stepping into/over, 284–285
debugging code
debuggers
the first bug, 287
How To, 285–286
tracing recursive methods,
598–599
Worked Example, 287
decision trees, 770
decisions. See also comparisons; if
statements.
Boolean operations, 209–214
decision trees, 182–183
declaring
array lists, 344–345
arrays, 308–311
classes, 11
constants, 134
instance variables, 80–81, 106
interface types, 467–468
methods, 45
variables, 34–36, 40–41
decrement, -- (minus signs), 138
default methods
conflicting, 474–475
description, 473–474
full code example, 474
default package, 401
definite loops, 250
delete method, java.nio.files.Files
class, A-23
Denver's luggage handling system, 192
dependencies
classes, 378–379, 381–382, 569–570
methods, 378–379, 381–382
depth-first search, 787–789
descendants, 767
Deutsch, L. Peter, 609
dialog boxes
file choosing, 523–524
full code example, 160, 524
for input/output, 160–161
showInputDialog method, javax.
swing.JOptionPane
class, A-35
showMessageDialog method, javax.
swing.JOptionPane
class, A-35
showOpenDialog method, javax.
swing.JFileChooser
class, A-34
showSaveDialog method, javax.
swing.JFileChooser
class, A-34
diamond syntax, 352
dice. See rolling dice.
dice throws, simulating, 279–280
Die.java class, 279–280
DieSimulator.java class, 280
Difference Engine, 658
Dijkstra, Edsger, 206
dimension constructor, A-13
directories, 9–10. See also files.
discovering classes, 376–377,
566–567, 576–578
distinct method, java.util.stream.
Stream<T> interface, 853, A-32
divide method, java.math.BigInteger
class, A-22
“Division by zero” errors, 14
do loops, 258
documentation
API (application programming
interface), 50–53
comments, 88–90
converting comments to. See
javadoc program
online help, 53
Swing, 911–918
documenting, methods, 579–581
dollar sign ($), in variable names, 37
dongles, 249
dot (.), name syntax, 403
double constructor, A-18
double data type
definition, 130t
floating-point numbers, 36
for financial calculations, 132
overflow, 131
precision, 131
storing in streams. See java.util.
stream.DoubleStream
interface.
double-black problem, 794–796
doubleValue method, java.lang.Double
class, A-18
doubly-linked lists, 684, (Worked
Example) 736
Dr. Java environment, 54
draw method, java.awt.Graphics
2D
class, 62–63, A-13
drawing. See also graphical
applications; specific shapes.
a car, 110–113
circles, 64–65
colors, 66
on a component, 60–63
ellipses, 64–65
a face, 67–68
fills, 66
a flag (How To), 114–117
graphical shapes (How To),
114–117
lines, 65
rectangles, 59–63
shape classes, 110–114
drawLine method, java.awt.Graphics
2D
class, 65, A-13
drawString method, java.awt.
Graphics2D
class, 65, A-13
dynamic method lookup, 440
E
E constant, java.lang.Math
class, 133
earthquake descriptions
full code example, 195
Loma Prieta quake, 193
Richter scale, 193
ElevatorSimulation2.java
class, 215–216
ElevatorSimulation.java class, 179
Ellipse2D.Double constructor, A-15
ellipses, drawing, 64–65
else statements, dangling else
problem, 201
empty method, java.util.Optional<T>
class, A-29
empty string versus null reference,
187–189
empty strings, 154
empty trees, 769
EmptyFrameViewer.java class, 60
encryption algorithms, 539
enhanced for loop. See also for
loops. array lists, 345–346
iterating through arrays, 317–318
ENIAC (electronic numerical
integrator and computer), 5
enumeration types, 203. See also
arrays.
EOFException, A-15
“equal exit cost” rule, 791–792
equal sign (=), assignment operator, 38–39
equal signs (==), relational operator
comparing object references, 187
comparing strings, 186, 189–190
equality testing, 184–185
versus equals method, 186
syntax, 184t
testing for null, 187–189
equality testing. See equal signs (==),
relational operator; equals
method.
equals method
comparing objects, 452–453
equals method (continued)
  inheritance, 456
  java.awt.Rectangle class, 187
  java.lang.Object class, 452, A-20
  java.lang.String class, 186–187, A-21
equalsIgnoreCase method, java.lang.String class, A-21
erasing type parameters, 835–837
error handling, input errors, 549–554. See also exception handling.
error messages
  logging, 208–209
  reading exception reports, 160
  stack trace, 160
  trace messages, 208–209
errors. See also specific errors.
  arrays, 314
  compile-time, 14
  dangling else problem, 201
  declaring constructors as void, 90
  declaring instance variables in local variables, 106
  diagnosing with encapsulation, 83
  “Division by zero,” 14
  exceptions, 14
  full code example, 14
  ignoring parameter values, 96
  logic, 14
  in loops, 243–245
  misspelling words, 15
  roundoff, 132, 185
  run-time, 14
  syntax, 14
  unbalanced parentheses, 142
  uninitialized object references, 107
  unintended integer division, 142
  unnecessary instance variables, 106–107
ESA (European Space Agency), 554
escape sequences, 156
EtchedBorder constructor, A-36
European Computer Manufacturers Association (ECMA), 686
evaluate method, javax.xml.xpath.XPath interface, A-36
Evaluatror.java class, 617–618
event handling, user-interface events, 491
event listeners
calling listener methods, 495
containers as, 499
  definition, 491
  event adapters, 506–507
  event source, 491
  forgetting to attach, 498
  full code example, 506
  inner classes for, 493–495
  keyboard events, 506
  mouse events, 502–505
  timer events, 499–502
  event-controlled loops, 250
  events
    button press, 496–498
    definition, 491
    timer, 499–502
  exception handlers, 542–543
  exception handling. See also error handling.
    application input errors, 549–554
    catch clause, 542–543, 548
    catching exceptions, 542–543
    checked exceptions, 543–545
    closing resources, 545–546
    definition, 540
    designing exception types, 546–548
    finally statement, 549
    internal errors, 543
    NoSuchElementException error, 860, 861
    NullPointerException error, 859
    sample application, 549–554
    squelching exceptions, 548
    throw early, catch late, 548
    throwing exceptions, 540–541, 548
    throws clause, 544–545
    try blocks, 542–543
    try statement, 542–543
    try/finally statement, 549
    try-with-resources statement, 545–546
    unchecked exceptions, 543
    exception handling, full code examples
      catching exceptions, 543
      checked exceptions, 544
      closing resources, 546
    exception reports, 160
    exceptions, definition, 14
    exclamation point, equal (!=), relational operator, 184
    exclamation point (!), not operator, 211
executable algorithms, 16
execute method
  java.sql.PreparedStatement interface, A-24
  java.sql.Statement interface, A-25
executeQuery method
  java.sql.PreparedStatement interface, A-24
  java.sql.Statement interface, A-25
executeUpdate method
  java.sql.PreparedStatement interface, A-24
  java.sql.Statement interface, A-25
exists method
  java.io.File class, A-15
  java.nio.files.Files class, A-23
exit method, java.lang.System class, A-21
exp method, java.lang.Math class, A-19–20
explicit parameters, 108
expression trees, 771
ExpressionCalculator.java class, 619–620
expressions, 137, 143
ExpressionTokenizer.java class, 618–619
extending classes. See inheritance.
extends reserved word, 429
extracting values from elements, 859

F
face, drawing, 67–68
FaceComponent.java class, 67–68
Fibonacci sequence, 604–609
FIFO (first in, first out), 699
Fifth-Generation Project, 217
file choosers, 523–525
File class, 520
File constructor, A-15
file dialog boxes
  choosing files from a list, 523–524
  full code example, 524
file names
  as string literals, 523
to paths, 849
FileInputStream constructor, A-16
FileNotFoundException, 521–522, 543
FileOutputStream constructor, A-16
files. See also folders.
  backing up, 10
definition, 9
buckets, 748

collisions, 747
efficiency, 751

finding elements, 749

hash codes, 747

iterating over hash tables, 750–755

linear probing, 755–756

open addressing, 748, 755–756

probing sequence, 755–756

sample code, 751–755

separate chaining, 748

`
hashCode`

method, java.util.Objects

class, 687, A-29

HashMap

constructor, java.util.

HashMap<K, V>

class, A-28

HashSet

constructor, A-28

HashSetDemo.java

class, 754–755

HashSet.java

class, 751–754

hasNext

method

advancing an iterator, 726

java.util.Iterator<E>

interface, 688–689, A-28

java.util.Scanner

class, A-30

hasNextDouble

method, java.util.

Scanner

class, 215, A-30

hasNextInt

method, java.util.Scanner

class, 214–215, A-30

hasNextLine

method, java.util.

Scanner

class, A-30

hasPrevious

method, java.util.

ListIterator<E>

interface, 684, A-28

HeapDemo.java

class, 806–807

heaps

adding nodes, 798–799
definition, 797
removing nodes, 800–801
sample code, 802–807
storing in arrays, 802
heapsort algorithm, 808–812

HeapSorter.java, 810–812

height of trees, 767

“Hello, World” program

analyzing, 11–13
source code, 11
writing, 8–10

HelloPrinter.java

class, 11

help, online, 53. See also
documentation.

higher-order functions

and comparators, 859

lambda expressions, 858–859

high-level languages, 6

history of computers

Altair 8800, 407
Apple II, 407
Babbage’s Difference Engine, 658
corporate monopolies, 58
first kit computer, 407
first programmer, 658
hardware evolution, 5
IBM, 58
microprocessors, 407
Microsoft, 58
personal computing, 407
standardization, 686
Turing machine, 612
Univac Corporation, 58
VisiCalc, 407
Hoff, Marcian E., 407

Houston, Frank, 355
Huffman trees, 771, (Worked Example) 774
IBM, history of computers, 58

IETF (Internet Engineering Task Force), 686

if statements. See also switch statements.

dangling else problem, 201
definition, 178
duplicate code in branches, 183
ending with a semicolon, 182
flowchart for, 179
input validation, 214–216
multiple alternatives, 193–196
nesting, 196–199
sample program, 179
syntax, 180

ifPresent method

java.util.Optional<T>

class, A-29

Optional type, 860–861

IllegalArgumentCause, java.lang.

IllegalArgumentException

540–541, A-18

ImageIcon

constructor, A-33

immutable classes, 379–380

implementation, classes. See classes, implementing.

implements reserved word, 469–471

implicit parameters, 108

import directive, 395

importing
classes from packages, 52
packages, 401

in object, java.lang.System

class, 145, A-21

income tax computation, 196–199

income tax rate schedule, 197

increment, ++ (plus signs), 138

indefinite loops, 250

indenting code
coding guidelines, A-44

with tabs, 182

IndexOutOfBoundsException, 543–544

infinite loops, 244

infinite recursion, 598

infinite streams, 851–852

INFO message level, 209

info method, java.util.logging.

Logger class, 209, A-31

information hiding. See encapsulation.
inheritance. See also polymorphism.
class relationships, 571–572
definition, 424
equals method, 456
full code example, 431, 449
generic classes, 833
generic methods, 833
indicating, 429

versus interfaces, 471–472

purpose of, 428

super reserved word, 437–438

inheritance hierarchy
developing (How To), 445–449

Worked Example, 450

initialization

arrays, 308–310

static methods, 395

initialization, variables

assignment statements, 38
definition, 35
local, 105
static, 395

Initials.java

class, 158

inner classes
definition, 488
drawing an ellipse, 65

as event listeners, 493–495

instance variables, 494

local variables, 494

Node class, 722–723

overview, 487–488

inorder traversal, 784–785
input. See also java.util.Scanner class; output.
  from a console, 155
definition, 4
dialog boxes, 160–161
  with dialog boxes (full code example), 160
from a keyboard, 145–146
prompting for, 145–146
reading, 145–146
redirecting, 262
strings, 155
input statements, syntax, 145
input validation, 214–216
InputStreamReader constructor, A-16
insertion sort
description, 645–646
full code example, 646
Worked Example, 667
instance methods, 143–144
instance variables
  access specifiers, 81
declaring, 80–81
declaring in local variables, 106
definition, 80
capsulation, 82–84
implementing classes, 91
return values, 82
syntax, 81
type, specifying, 81
unnecessary, 106–107
instanceof operator
description, 453–455
full code example, 454
instantiating, interface types, 472
int data type, 36, 130t
int values, storing in streams. See
  IntStream interface.
integer division, 138–139, 142
Integer constructor, A-22
integer values, mixing with floating-point, 137
integers
  computation overflow, 131
definition, 35
integrated development environment, 8
Intel Corporation, 144
interface types
  constants, declaring, 473
declaring, 467–468
definition, 466
  discovering, 466–467
generic, 486–487
implementing, 469–471
  implements reserved word, 469–471
instantiating, 472
public methods, 472
interface variables
casting from interfaces to classes, 476–477
converting from classes, 475
full code example, 477
invoking methods on, 476
interfaces. See also specific interfaces.
  for callbacks, 482–486
  functional, 485–486
  versus inheritance, 471
  modifying method parameter types, 495
  static methods, 473
  Worked Example: number sequences, 477
interior nodes, 767
international alphabets, 161
International Organization for Standardization (ISO), 686
Internet, history of, 456
Internet Engineering Task Force (IETF), standardization of computers, 686
Internet Explorer, 456
Java language. See also applets; programming.
description, 6–7
high-level languages, 6
integrated development environment, 8
portability, 7
versions, 6t
Java library, 6–7, A-13–38. See also packages; specific elements.
  inheritance hierarchy, A-9–12.
Java operators, summary of, A-5–6t
Java programs. See also specific elements.
class files, 9
compilation process, 9
compilers, 6
getting started. See “Hello, World” program.
machine code, 6
source code, 9
source files, 9
java.awt package, A-12–14
implementing stacks as, 741–743
list iterators, 683–685
nodes, 681–682
sample code, 685
structure of, 681–682
linked lists, implementing
adding/removing elements, 723–724, 726–729
advancing an iterator, 725–726
doubly-linked lists (Worked Example), 736
efficiency, 729–732, 732t
full code example, 735
inner classes, 722–723
Iterator class, 724–725
LinkedListIterator class, 724–725, 736
Node class, 722–723
sample code, 732–736
setting element values, 729
static classes, 736
LinkedList constructor, java.util.
Iterator<E> interface, A-28
LinkedList class, 682–683
LinkedListIterator class, 724–725
LinkedListStack java class, 732–735
ListDemo java class, 693–694
listeners. See event listeners
ListIterator class, 724–725
ListIterator method
java.util.LinkedList<E> class, 683
java.util.List<E> interface, A-32
ListIterator java class, 735–736
lists, creating, 678–679
literals, string, 154
load method, java.util.Properties class, A-29
local variables
declaring instance variables in, 106
definition, 105
duplicate names, 202
full code example, 105
garbage collection, 105
initializing, 105
lock method, java.util.concurrent.
locks.Lock interface, A-30
log method, java.lang.Math class, 140t, A-19–20
log10 method, java.lang.Math class, 140t, A-19–20
logging messages, 208–209
logic errors, 14
Loma Prieta earthquake, 193
long data type, 130, 131
long values, storing in streams. See LongStream interface.
LongStream interface, 863
loop and a half problem, 262–263
loopFib.java class, 607–608
loops
asymmetric bounds, 256
Boolean variables, 261
bounds, choosing, 256
break statements, 263–264
common errors, 243–245
count-controlled, 250
counting iterations, 256
credit card processing (Worked Example), 275
definite, 250
definition, 238
do loop, 258
enhanced for loop, 317–318. See also for loops.
event-controlled, 250
flowcharting, 259
full code example, 258, 317–318
hand tracing, 245–249
improving recursion efficiency, 608–609
indefinite, 250
infinite, 244
loop and a half problem, 262–263
for loops, 250–255, 257. See also enhanced for loops.
manipulating image pixels (Worked Example), 278
nesting, 275–278
off-by-one errors, 244–245
post-test, 258
pre-test, 258
redirecting input/output, 262
sentinel values, 259–262
while, 238–243, 255
writing (How To), 272–275
loops, common algorithms
averages, computing, 268
comparing adjacent values, 271–272
counters, 268–269
finding first match, 269
full code example, 271
maximum/minimum values, finding, 270
prompting for first match, 270
totals, computing, 268
luggage handling system, 192
M
machine code, 6
magic numbers, 137
main method
command line arguments, 533–535
definition, 11
managing object properties, common class patterns, 388
map method, java.util.stream.
Stream<T> interface
description, A-32
as high-order function, 858
mapping a primitive-type stream, 864
transforming streams, 852
MapDemo java class, 693–694
mapping primitive streams, 864
maps
definition, 679–680
description, 692–694
updating, 695
mapToDouble method, java.util.
stream.Stream<T> interface, A-32
mapToInt method, java.util.stream.
Stream<T> interface, A-32
mapToLong method, java.util.stream.
Stream<T> interface, A-32
mapToObj method
java.util.stream.DoubleStream
interface, A-32
java.util.stream.IntStream
interface, A-32
java.util.stream.LongStream
interface, A-32
mathematical operations. See arithmetic operations.
method summary, java.lang.Math class, 140t
matrices. See arrays, two-dimensional.

max method, java.lang.Math class
description, A-19
Optional type, 861
primitive streams, 865
syntax, 140
as terminal operator, 862
maxBy method, java.util.stream.
Collections class, A-31
maximum values
finding in arrays, 319
finding with loops, 270
getting, 130
stream algorithms for, 869
streams, 866–868
MAX_VALUE constant, java.lang.Integer
class, 130
MeasurableTester.java
class, 470–471
Measurer.java
class, 483–484
MeasurerTester.java
class, 484–485
Medals.java
class, 340–341
menu bars, 905
menu items, 905
menus
code sample, 907–910
definition, 905
implementing a class for, 99
javax.swing.JMenu
class, 905–910
t, A-19
overview, 905–907
submenus, 905
merge method, 694
MergeSortDemo.java
class, 649
MergeSorter.java
class, 648–649
messages. See error messages.
method references, lambda
eexpressions, 857–858
methods. See also specific methods.
abstract, 443–444
accessing data without modifying.
See accessor methods.
accessor, 48–49
arguments, 42–43, 315–316
call-by-reference, 382–386
call-by-value, 382–386
calling, 12, 32–33
versus constructors, 85
declaring, 45
default, 473–475
definition, 11, 33
documenting, 579–581
duplicate names. See overlapping.
final, 444
functional interfaces, 485–486
versus functions, 858
implementing classes, 93–95
instance, 143–144
invoking on interface variables, 476
invoking on numbers, 143–144
measuring run time, 642
modifying data. See mutator
methods.
mutator, 48–49
naming conventions, 37
overloading, 45
passing information to. See
parameters; this reference.
private implementation, 42
public interfaces, classes, 84–85
public interfaces to classes, 41–42
return values, 43–44
static, 143–144
in UML diagrams, 573
verbs as, 566
with no results. See null, testing
for; Optional type.
methods, designing
accessors, 379–380
cohesion, 377–378, 381–382
consistency, 381–382
dependencies, 378–379, 381–382
mutators, 379–380
public interfaces, 377–378
side effects, 380–381
methods, full code examples
accessor methods, 49
method calls, 45
mutator methods, 49
methods, static
accessing instance variables, 394
definition, 391–393
full code example, 393
initialization, 395
min method, java.lang.Math class
description, 140t, A-19
primitive streams, 865
as terminal operator, 862
minBy method, java.util.stream.
Collections class, A-31
MinHeap.java
class, 802–806
minimum values
finding in arrays, 319, 329
finding with loops, 270
getting, 130
stream algorithms for, 869
streams, 866–868
minus sign (--), subtraction operator, 36
minus signs (–), decrement, 138
MIN_VALUE constant, java.lang.Integer
class, 130t, A-19
misspelling words, 15
mock objects, 489–490
mod method, java.math.BigInteger
class, A-22
modeling
moving objects, common class
patterns, 389–391
objects with distinct states,
common class patterns, 388–389
modulus, 138–139
Monte Carlo method, 281–282
MonteCarlo.java, 281–282
Morris, Robert, 316
Mosaic, 456
mouseClicked method, A-14
mouseEntered method, A-14
mouseExited method, A-14
mousePressed method, A-14
mouseReleased method, A-14
move method, java.nio.files.Files.Files
class, A-23
MoveTester.java
class, 54
movie database, analyzing, (Worked
Example), 874
multidimensional arrays, 343
multiplicities, 574
multiply method
java.math.BigInteger class, 136,
A-22
java.math.BigDecimal class, 136,
A-22
Murphy, Walter, 586
mutations in parallel streams, 863
mutator methods
definition, 379
Index

description, 48–49, 379–380
full code example, 48–49
Mycin program, 217

N
\nnewline character, 156
name clashes, packages, 401–402
naming conventions
classes, 37
constants, 133
constructors, 86
local variables, 202
methods, 37
packages, 401–402
test classes, 406
type variables in generic classes, 826
variables, 37, 41
Naughton, Patrick, 6
negative-red problem, 794–796
nesting
if statements, 196–199
loops, 275–278
panels, 885–886
networks, definition, 4
new newCondition method, java.util.
.concurrent.locks.Lock
interface, A-31
newDocument method, javax.xml.
parsers.DocumentBuilder class, A-36
newDocumentBuilder method, javax.xml.parsers.
DocumentBuilderFactory class, A-36
newInstance method, javax.xml.
parsers.DocumentBuilder class, A-36
newInstance method
javax.xml.parsers.
DocumentBuilderFactory class, A-36
javax.xml.xpath.XPathFactory
class, A-37
newXPath method, javax.xml.xpath.
XPathFactory class, A-37
next nextLine method, java.util.Scanner
class
description, A-30
reading lines, 527–528
Nicely, Thomas, 144
“no double reds” rule, 791–792
no-argument constructors, 86
Node class, 722–723, 778
nodes, 681–682, 767
noneMatch method, java.util.stream.
Stream<T> interface, 862, A-32
NoSuchElementException error, 860, 861
notify notify method, java.lang.Object
class, A-20
notifyAll method, java.lang.Object
class, A-20
nouns as classes, 566
null, testing for, 187–189. See also
Optional type.
null reference versus empty string,
187–189
NullPointerException error, 859
number literals, 131
number sequences (Worked
Example), 477
number variables versus object
variables, 56
numbers
big, 136
constants, 132–136
converting from strings, 528–529
data types, 130–132
with fractions. See floating-point
numbers.
without fractions. See int type.
invoking methods on, 143–144
magic, 137
primitive types, 130

O
Object class. See also superclasses.
definition, 450
equals method, 452–453
instanceof operator, 453–454
toString method, 450–451, 455
object construction
arguments, 47
definition, 47
invoking a constructor like a
method, 48
overview, 46–48
syntax, 47
object references
definition, 56
full code example, 57
number variables versus object
variables, 56
overview, 55–58
uninitialized, 107
object variables versus number
variables, 56
ObjectInputStream constructor, A-16
ObjectOutputStream constructor, A-16
objects
arrays of, 314–315
comparing, 665–667. See also
equals method.
copying, 477–481
definition, 32
hand tracing, 103–105
mock, 489–490
testing. See mock objects.
OFF message level, 209
test errors, 244–245
online help, 53. See also
documentation.
on open addressing, 748, 755–756
openStream method, java.net.URL
class, A-238
operators. See specific operators.
Optional type. See also null, testing for.
full code example, 860
get method, 861
Optional type (continued)
ifPresent method, 861
isPresent method, 861
obtaining the value of, 860
orElse method, 860–861
results without values, 861
orElse method
java.util.Optional<T> class, A-29
Optional type, 860–861
org.w3c.dom package, A-37
org.w3c.dom.Document interface, method summary, A-37. See also specific methods.
org.w3c.dom.DOMConfiguration interface, method summary, A-37. See also specific methods.
org.w3c.dom.DOMImplementation interface, method summary, A-37. See also specific methods.
org.w3c.dom.Element interface, method summary, A-37. See also specific methods.
org.w3c.dom.ls package, A-37
org.w3c.dom.ls.DOMImplementationLS interface, method summary, A-37. See also specific methods.
org.w3c.dom.ls.LSSerializer interface, method summary, A-37. See also specific methods.
org.w3c.dom.Text interface, A-37
out object, System class, 12–13, 33
output. See also input.
definition, 4
dialog boxes, 160–161
with dialog boxes (full code example), 160
format specifiers, 146–148
formatting, 146–148
line breaks, 156
redirecting, 262
overflow, computation
definition, 131
double data type, 131
overloading methods, 45, 603
accidentally, 437
overriding methods
forcing, 443–444
preventing, 444
superclass, 429, 433, 443–444
packages. See also specific packages.
 . (dot), name syntax, 403
accessing, 403–404
API documentation, 52–53
default, 401
definition, 400
description, 7
importing, 401
importing classes from, 52
Java library, 6–7
name clashes, 401–402
nesting, 401
programming with (How To), 404–405
source files, 402–403
summary of, 400
syntax, 401
paintComponent method, javax.swing.JComponent class, 61–63, A-34
PairDemo.java class, 828
Pair.java class, 827–828
palindromes, identifying (How To), 599–602
panels (user-interface)
creating, 496–498
definition, 884
nesting, 885–886
parallel arrays, 314–315
parallel method, java.util.stream.Stream<T> interface, A-33
parallel streams, 851, 863
@param tag, 88
parameter passing, 44
parameter variables
ignoring, 96
javadoc program, 88
parameters
explicit, 108
implicit, 108
this reference, 107–109
parent nodes, 767
parentheses ( ), balancing, 142, 701
in arithmetic operations, 137, 142
in object construction, 47
parse method, javax.xml.parsers.DocumentBuilder class, A-36
parseDouble method, java.lang.Double class, A-18
converting strings to numbers, 160, 528–529
parseInt method, java.lang.Integer class, A-18
converting strings to numbers, 160, 528–529
partially filled arrays, 312–313
PartialSolution.java class, 623–624
partitioning the range, 652–653
patent, definition, 539
paths, 767
paths from a filename, 849
PCs (personal computers), schematic design of, 4. See also computers.
peek method
java.util.Queue<E> interface, 699, A-29
java.util.Stack<E> class, 698
Pentium floating-point bug, 144
percent sign (%), modulus, 138–139
performance estimation, array algorithms
full code example, 663
linear time, 659–660
logarithmic time, 662–664
quadratic time, 660–661
triangle pattern, 661–662
performance estimation, big-Oh notation, 643–645
peripheral devices, 4. See also specific devices.
permutations of strings, 609–612
Permutations.java class, 610–611
personal computing, 407
PGP (Pretty Good Privacy) encryption, 539
PI constant, java.lang.Math class, 133, A-20
pictures, editing (Worked Example), 55. See also drawing; graphical applications.
piracy, digital, 249
pivot, 652
pixel images, manipulating with loops (Worked Example), 278
plus sign (+)
addition operator, 36
concatenation operator, 155
plus signs (++), increment, 138
Point2D.Double constructor, A-15
polymorphism. See also inheritance.
dynamic method lookup, 440, 442
overview, 439–440
pop method, `java.util.Stack<E>` class, 698
`PopulationDensity.java` class, 538
potability, 7
postorder traversal, 785–786
post-test loops, 258
pow method, `java.lang.Math` class, 140, A-19
powers, calculating, 139–140
`PowerTable.java` class, 276–277
precedence
  arithmetic operations, 137
  Boolean operators, 210
  Java operators, A-5–6
precision
  double data type, 131
  floating-point numbers, 131–132
predicate, streams, 853
preorder traversal, 785–786
prepareStatement method, `java.sql.Connection` interface, A-24
pre-test loops, 258
previous method, `java.util.ListIterator<E>` interface, A-28
primary storage, 3
primitive number types, 130
primitive streams. See streams, primitive.
print commands, full code example, 13
print method, 13
  `java.io.PrintStream` class, A-16
  `java.io.PrintWriter` class, 520–522, A-16
println method, 13
  `java.io.PrintStream` class, A-16
  `java.io.PrintWriter` class, 520–522, A-16
printStackTrace method, `java.lang.Throwable` class, A-22
`PrintStream` constructor, A-16
`PrintWriter` constructor, A-16
priority queues
  definition, 679
  description, 699–700
  full code example, 700
`PriorityQueue` constructor, A-29
privacy issues, databases, 586
private method implementation, 42
probing sequence, 755–756
problem statements, converting to pseudocode (How To), 149–152
`Product.java` class, 584–585
program comparator, 612–613
program development, examples. See “Hello, World” program; printing invoices.
programming. See also applets; Java language.
  compilers, 6
  definition, 2
  getting started. See “Hello, World” program.
  high-level languages, 6
  machine code, 6
  scheduling time for, 208
  solving simpler problems first, 395–399
prompting
  for first match, with loops, 270
  for input, 145–146
protected access feature, 444–445
pseudocode
  for algorithms, 19
  definition, 17
  description, 18
  writing (How To), 149–152
`Random` constructor, A-29
random numbers
  finding approximate solutions, 281–282
  generating, 279–280
  Monte Carlo method, 281–282
  pseudorandom numbers, 280
`RandomAccessFile` class, method summary, A-17
`RandomAccessFile` constructor, A-17
radiation therapy incidents, 355
radio buttons
  borders, 895
  definition, 894–895
  grouping, 894–895
`Random` constructor, A-29
random numbers
  finding approximate solutions, 281–282
  generating, 279–280
  Monte Carlo method, 281–282
  pseudorandom numbers, 280
`RandomAccessFile` class, method summary, A-17
`RandomAccessFile` constructor, A-17
readInt method, java.io.RandomAccessFile class, A-17
readObject method, java.io.ObjectInputStream class, A-16
Rectangle constructor, A-13
RectangleComponent2. java class, 503–504
RectangleComponent. java class, 62–63, 500–501
RectangleFrame2. java class, 504–505
RectangleFrame. java class, 501
rectangles
comparing, 187
drawing, 59–63
moving (full code example), 506
RectangleViewer2. java class, 505
RectangleViewer. java class, 62–63, 501–502
recursion
backtracking, 620–626
changing to a loop, 608–609
debugging, 598–599
definition, 594
finding files (Worked Example), 602
full code example, 602
identifying a palindrome (How To), 599–602
infinite, 598
permutations, 609–612
setting breakpoints, 598–599
tracing, 598–599, 605–607
triangle numbers, 594–598
recursion, efficiency
changing to a loop, 608–609
Fibonacci sequence, 604–609
full code example, 609
trace messages, 605–607
recursion, mutual
description, 614–616
sample code, 617–620
syntax diagrams, 614
tokens, 615
recursive helper methods
description, 602–604
full code example, 603
overloading, 603
recursive methods, call pattern, 595–596
RecursiveFib. java class, 604–605
RecursiveFibTracer. java class, 605–607
red-black trees
basic properties, 790–792
definition, 790
double-black problem, 794–796
efficiency, 796r
“equal exit cost” rule, 791–792
implementing (How To), 796
inserting nodes, 792–793
negative-red problem, 793–796
“no double red” rule, 791–792
removing nodes, 793–796
redirecting input/output, 262
ReentrantLock constructor, A-31
reflection, 838–839
regression testing, 352–354
regular expressions, 532
relational operators, 184–185, 212
remainders, calculating, 138–139
remove method
java.util.ArrayList<E> class, 345, A-26
java.util.Collection<E> interface, A-27
java.util.Map<K, V> interface, A-29
java.util.PriorityQueue<E> class, A-29
java.util.Queue<E> interface, 699
removeFirst method, java.util.LinkedList<E> class, 726–727, A-28
removeLast method, java.util.LinkedList<E> class, A-28
repainting graphic components, 502
replace method, java.lang.String class, 44, A-21
replaceAll method, java.lang.String class, 532, A-21
requirements, gathering, 575–576
reserved words
summary of, A-7–8r
as illegal variable names, 37
resizing arrays, 311, 312–313
resolve method, java.nio.file.Path interface, A-23
return statement, 82
@return tag, 88
return values, 43, 82
reusing algorithms, 327
reverse Polish notation, 702–703, 709
Richter scale, 193r
Rivest, Ron, 539
rocket incident, Ariane, 554
rollback method, java.sql.Connection interface, A-24
rolling dice
simulating, 279–280
Worked Example, 332
root nodes, 767
roots, calculating, 139–140
round method, java.lang.Math class
definition, 140r
description, A-20
rounding floating-point numbers, 140
rounding
description, 131–132
full code example, 140
roundoff errors, 132, 185
RSA encryption, 539
Rumbaugh, James, 378
run method, java.lang.Runnable
interface, A-20
run-time errors, 14
run-time stacks, 698

safety, Java, 7
sales tax calculation, full code
example, 104
Scanner class
character encoding, 525
constructing with a string, 523
reading text files, 520–522
Scanner constructor, A-30
scheduling time for programming,
208
ScoreTester.java class, 353
scroll bars, 892
searching
binary, 655–658
breadth-first, 787–789
depth-first, 787–789
linear, 654–655
secondary storage, 3–4
seek method, RandomAccessFile
class, A-17
SelectionSortDemo.java class, 638
SelectionSorter.java class, 637–638
SelectionSortTimer.java class, 641
semicolon (;)
ending if statements, 182
ending method statements, 11
omitting, 13
sentinel values, 259–262
SentinelDemo.java class, 260–261
separate chaining, 748
sequential search, 320
ServerSocket constructor, A-22
set method
java.util.ArrayList<E> class, 344–345, A-26
java.util.ListIterator<E>
interface, 684, 729, A-28
setAttribute method, org.w3c.dom.
Element interface, A-37
setAutoCommit method, java.sql.
Connection interface, A-24
setBorder method, java.awt.
JComponent class, 895, A-34
setColor method, java.awt.Graphics
class, 66, A-13
setDefaultCloseOperation method,
javax.swing.JFrame class, 59,
A-34
setDouble method, java.sql.
PreparedStatement interface,
A-24
setEditable method
javax.swing.JComboBox class, A-33
javax.swing.text.JTextComponent
class, 891, A-36
setFocusable method, java.awt.
Component class, 506, A-12
setFont method, java.awt.
Component class, A-34
setIfModifiedSince method, java.net.
URLConnection class, A-23
setIgnoringElementContentWhitespace
method, javax.xml.parsers.
DocumentBuilderFactory class,
A-36
setInt method, java.sql.
PreparedStatement interface,
A-24
setMenuBar method, java.awt.
JFrame class, 905–910, A-34
setLayout method, java.awt.
Container class, 884–885, A-13
setLevel method, java.awt.
Logger interface, 209
setLine method, java.awt.
geom.Line2D class, A-15
setLocation method
java.awt.geom.Point2D class, A-15
java.awt.Rectangle class, A-14
setParameter method, org.w3c.dom.
DOMConfiguration interface,
A-37
setPreferredSize method, java.awt.
Component class, 887, A-12
sets
adding/removing elements, 688
binary search trees, 687
choosing an implementation,
687–688
definition, 679
description, 687–689
hash tables, 687
iterating over, 688–689
listing elements, 688–689
setSelected method, java.awt.
AbstractButton class, A-33
setSelectedItem method, java.awt.
JComboBox class, A-34
setSize method
java.awt.Component class, 59, A-12
java.awt.Rectangle class, 51–52,
A-14
setText method, java.awt.
Component class, 59, A-36
setTitle method, java.awt.
Frame class, 59, A-13
setValidating method, java.awt.
parsers.DocumentBuilderFactory
class, A-36
setVisible method, java.awt.
Component class, 59, A-12
Shamir, Adi, 539
shape classes, 110–114
shell scripts, 354
shipping costs, computing (full code
example), 204–205
short circuiting Boolean evaluation,
213
short data type, 130
showInputDialog method, java.awt.
JOptionPane class, 160, A-35
showMessageDialog method, java.awt.
JOptionPane class, 161, A-35
showOpenDialog method, java.awt.
JFileChooser class, 524, A-34
showSaveDialog method, java.awt.
JFileChooser class, 524, A-34
sibling nodes, 767
side effects, 380–381
signal method, java.util.concurrent.
locks.Condition interface, A-30
signalAll method, java.util.
concurrent.locks.Condition
interface, A-30
simulation programs, 279–282
sin method, java.lang.Math class,
140, A-20
single-step command, 284
size method
java.util.ArrayList<E> class, 344,
A-26
java.util.Collection<E>
interface, A-27
skip method, java.util.stream.
Stream<T> interface, A-33
slash, asterisks (**), comment indicator, 88
slash (/), division operator, 36, 137
slash asterisk... (**...**), comment indicator, 88
slash (/), division operator, 36, 137
slashes (//), comment delimiter, 38
sleep method, java.lang.Thread class, A-21
Socket constructor, A-22
software, definition, 2. See also programming.
software piracy, 249
sort method
java.util.Arrays class, 327, 664–665, A-26
java.util.Collections class, 664–665, A-27
sorted method
lambda expressions, 855–856
transforming streams, 853
sorting arrays, 327
new streams, 853
objects, with lambda expressions, 855–856
sorting algorithms
definition, 636
full code example, 646, 653
insertion sort, 645–646
insertion sort (Worked Example), 667
partitioning the range, 652
pivot, 652
quicksort, 652–653
sorting algorithms, merge sort
description, 647
full code example, 652
measuring performance of, 650–652
sample programs, 648–649
sorting algorithms, selection sort
description, 636–637
measuring performance of, 642–646
sample programs, 637–639, 640–641
source code
“Hello, World” program, 11
Java programs, 9
source files, 9
spaces in
comparisons, 186
expressions, 143
input. See white space.
variable names, 37
spaghetti code, 204–205
spell check, sample code, 690–691
spelling errors, 15
split method, java.lang.String class, 532, A-21
sqrt method, java.lang.Math class, 140t, A-20
stack trace, 160
stacks
algebraic expressions, 703–706
backtracking, 706–708
balancing parentheses, 701
definition, 679
description, 698
full code example, 700, 708
LIFO (last in, first out), 698
reverse Polish notation, 702–703
run-time, 698
stacks, implementing
as arrays, 743
as linked lists, 741–743
sample code, 742–743
standardization of computers, 686
start method
java.lang.Thread class, A-22
javax.swing.Timer class, 500–501, A-40
statements
definition, 11
punctuating, 11, 13
static classes, 736
static imports, 394–395
static methods
accessing instance variables, 394
definition, 392–393
full code example, 393
initialization, 395
in interfaces, 473
minimizing use of, 393–394
overview, 143–144
static imports, 395
stepping through code, 283–285
stop method, javax.swing.Timer class, A-35
stopwatch program, 639–642
StopWatch.java class, 640–641
storage devices. See also specific devices.
primary storage, 3
secondary storage, 3–4
storyboards, 265–267. See also flowcharts.
stream library, 846
stream method, java.util.Collection<E> interface, 849, A-27
StreamDemo.java, 847–848, 853–854
Stream.of method, 847–848
streams
coding for readability, 851
versus collections, 847
counting words, 846
definition, 847
How To, 871–874
with lambda expressions, 846
lazy processing, 847
picking out words, 846
sample code, 847–848
storing int, long, and double. See streams, primitive.
Worked Examples, 874
streams, collecting results from collect method, 850
Collectors.joining method, 850
combining strings, 850–851
converting to arrays or collections, 850–851
extracting values from elements, 859
full code example, 850
limiting results, 852
overview, 850–851
toArray method, 850
streams, common algorithms
averages, 869
comparing adjacent values, 870
counting matches, 869
element separators, 869–870
filling an array, 868–869
full code example, 868
linear search, 870
maximum values, 869
minimum values, 869
sums, 869
streams, grouping results
averages, 866–868
averagingDouble collector, 867
averagingInt collector, 867
averagingLong collector, 867
collectors, 866
counts, 866–868
GroupDemo.java class, 867–868
maximum values, 866–868
minimum values, 866–868
overview, 866–867
sample code, 867–868
summingDouble collector, 866–868
summingInt collector, 866–868
summingLong collector, 866–868
sums, 866–868
streams, primitive
average method, 865
of consecutive integers, 863–864
creating, 864
doubleStream interface, 863
toIntStream method, 863–864
toLongStream interface, 863
mapping, 864
max method, 865
min method, 865
processing, 864–865
sum method, 865
streams, producing
from arrays, 847–848
Files.lines method, 849
callable, 849
generate method, 851
infinite, 851–852
iterate method, 851
making a path from a file name, 849
mutations in parallel streams, 863
from old streams, 847, 853
parallel streams, 851, 863
stream method, 849
Stream.of method, 847–848
from a terminated stream, 855
strings. See also characters; substrings.
versus characters, 156–157
strings. See also characters; substrings.
versus characters, 156–157
sum method, primitive streams, 865
summing
array elements, 319
with loops, 268
stream algorithms, 869
terminal operations (continued)  
min method, 862  
noneMatch method, 862  
terminated streams, producing new streams, 855  
terminating algorithms, 16  
test cases, selecting, 206–207  
test method, java.util.function.  
Predicate &lt;T&gt;  
interface, A-31  
terminated streams, producing new streams, 855  
test suites, 352  
tester classes, 100–101  
testing  
black-box, 206  
boundary cases, 206–207  
classes, 54–55  
code coverage, 206  
for null, 187–189  
mock objects, 489–490  
programs, 53–54  
regression, 352–354  
unit, 100–101  
white-box, 206  
text areas, 891–893  
text fields, 888–890  
text files, reading and writing. See also reading input; writing output.  
How To, 536–538  
overview, 520–522  
text I/O  
code sample, 889–890, 892–893  
multiple lines, 891–893  
scroll bars, 892  
single line, 888–890  
text areas, 891–893  
text fields, 888–890  
thenComparing method, java.util.  
Comparator&lt;T&gt;  
interface, 859, A-27  
therac-25 incidents, 355  
this reference  
description, 107–109  
full code example, 109  
throw early, catch late, 548  
throw statement, 540–541  
Throw&One constructor, A-22  
throwing exceptions, 540–541, 548, 549  
throws clause, 544–545  
tiles for flooring, computing by hand, 152–154  
programmatically (full code example), 153  
tiling a floor, algorithm for (Worked Example), 21–22  
timer events, 499–502  
Timer constructor, A-35  
TitleBorder method, javax.swing.  
border.EtchedBorder class, A-36  
toArray method, java.util.stream.  
Stream&lt;T&gt;  
interface, 850, A-33  
toDegrees method, java.lang.Math  
class, 140r, A-20  
tokens, 615  
toList method, java.util.stream.  
Collectors class, A-31  
toLowerCase method, java.lang.String  
class, 41–42, A-21  
toRadians method, java.lang.Math  
class, 140r, A-20  
toSet method, java.util.stream.  
Collectors class, 866, A-31  
toString method  
full code example, 454  
inheritance, 455  
inserting element separators, 319–320  
java.lang.Integer class, A-19  
java.lang.Object class, A-20  
java.util.Arrays class, A-26  
overriding, 450–451  
Total.java class, 521–522  
totals, computing with loops, 268  
toUpperCase method, java.lang.String  
class, 41–42, A-21  
Towers of Hanoi (Worked Example), 626  
trace messages, 208–209  
tracing. See hand-tracing.  
tracing recursive methods, 598–599, 605–607  
transforming streams. See streams, transforming.  
transistors, 3  
translate method, java.awt.Rectangle  
class, 49, 53–54, A-14  
travel time computation (Worked Example), 154  
tree concepts  
ancestors, 767  
child nodes, 767  
computing the size, 769  
descendants, 767  
empty trees, 769  
full code example, 769  
height, 767  
implementing a tree, 768  
interior nodes, 767  
leaf nodes, 767  
nodes, 767  
overview, 766  
parent nodes, 767  
paths, 767  
root nodes, 767  
sibling nodes, 767  
subtrees, 767  
tree iterators, 789  
tree traversal  
breadth-first search, 787–789  
dept-first search, 787–789  
full code example, 789  
inorder, 784–785  
postorder, 785–786  
preorder, 785–786  
tree iterators, 789  
visitor pattern, 786–787  
TreeMap constructor, A-30  
TreeSet constructor, A-30  
triangle numbers, 594–598  
Triangle.java class, 596–597  
TriangleTester. java class, 597  
trim method, 528  
trimming white space, 528  
try block, 542–543  
try statement, 542–543  
try/finally statement, 549  
Turing, Alan, 612–613  
Turing machine, 612–613  
two-dimensional arrays. See arrays, two-dimensional.  
type parameters  
erasing, 835–837  
full code example, 837  
in static context, 838  
type parameters, constraining  
full code example, 835  
generic classes, 831–832  
generic methods, 831–832  
wildcard types, 834–835  
type variables  
definition, 824  
naming, 826  
types of data. See data types.
U

UML (Unified Modeling Language)
attributes, 573
definition, 378
methods, 573
in program design (How To), 572–573
relationship symbols, 572
tsample program, 578–579
The Unified Modeling Language
User Guide, 575
unambiguous algorithms, 16
unchecked exceptions, 543
undeclared variables, 40
underscore (_, in variable names, 37
unfilled arrays, 314
Unicode, 156–157, 161
The Unified Modeling Language
User Guide, 575
uninitialized arrays, 314
uninitialized variables, 38, 40
unintended integer division, 142
union method, java.awt.Rectangle
class, A-14
unit testing
annotating test methods, 405–406
definition, 100
JUnit framework, 405–406
naming test classes, 406
tester classes, 100–101
Univac Corporation, history of
computers, 58
unlock method, java.util.concurrent.
locks.Lock interface, A-31
URL constructor, A-23
URLConnection constructor, A-23
useDelimiter method, java.util.
Scanner class
description, A-30
reading characters, 526
user-interface components. See also
specific components.
bbuttons, 496–498
lables, 496
panels, 496–498
repainting, 502
UTF-8 encoding, 525
utility classes, 376
V
validating numeric input, 529
variables. See also instance variables;
local variables.
assigning values to, 38–39
comments, 38
declaration versus assignment, 40–41
declaring, 34–36
definition, 35
full code example, 39
initializing, 35, 38
inspecting during debugging, 283–285
in for loop headers, 257
naming conventions, 37, 41
object versus number, 56
scope, minimizing, 201–202
syntax, 35
undeclared, 40
uninitialized, 38, 40
variables, static
definition, 391–393
full code example, 393
initialization, 395
static imports, 395
verbs as methods, 566
vertical bars (|), or operator, 209–210, 212, 213
virtual machine, definition, 7
VisiCalc, 407
visitor pattern, 786–787
void reserved word
declaring constructors, 90
definition, 45
volume of a pyramid, computing
(Worked Example), 152
Volume.java class, 148
voting machines, 102
W
W3C (World Wide Web
Consortium), 686
wait method, java.lang.Object class,
A-20
walk method, java.nio.files.Files
class, A-23
web pages, reading (full code
example), 523
while loops, 238–243, 255
while statement, syntax, 239
white space
consuming, 525–526
trimming, 528
white-box testing, 206
wildcard types, type parameters,
834–835
wildcards, full code example, 835
Wilkes, Maurice, 287
word frequency, counting (Worked
Example), 695
words
counting, 846
picking out, 846
reading, 525–526
WorkOrder.java class, 806
world population table (Worked
Example), 341
World Wide Web, 456
wrapper classes, 347, 347
write method, java.io.OutputStream
class, A-16
writeChar method, RandomAccessFile
class, A-17
writeChars method, RandomAccessFile
class, A-17
writeDouble method, RandomAccessFile
class, A-17
writeInt method, RandomAccessFile
class, A-17
writeObject method, java.
io.ObjectOutputStream class,
A-16
writeToString method, org.w3c.dom.
LSerializer interface, A-37
writing output, 530–532. See also
reading input; text files,
reading and writing.
writing text files, 520–522
Z
Zimmerman, Phil, 539