Index

Note: Page numbers in **boldface** refer to definitions of key terms.

Special Characters

& (ampersand), **259**, 259–261, 262–265, 269
<> (angle brackets), 39, 68, 359
* (asterisk), 32, 92, 93, 218, 312, 1030
@ (at sign), 1030
\ (backslash), 73, 677, 688
: (colon), 276–277, 279
{} (curly braces), 17, 21, 39
$ (dollar sign), 1016
“ (double quotation mark), 73
= (equal sign), 53, 68, 244, 245, 246, 247
! (exclamation point), 68, 247, 277
/ (forward slash), 32, 92, 93, 312, 677, 1030
(hash or pound sign, octothorpe), 1017
- (minus sign), 92, 93, 312, 501
() (parentheses), 39, 123–124
% (percent sign), 92, 93, 312, 1013, 1024
| (pipe), **261**, 262, 269
+ (plus sign), 57, 92, 93, 312, 365, 501
? (question mark), 276–277, 279
‘ (single quotation mark), 73
[] (square brackets), 39

A

`abs(x)` method, 216
absolute paths, **679**
  converting relative paths to, 680–681
abstract classes, **548**, 548–557
abstract data types (ADTs), **160**
`abstract` keyword, 549
abstract methods, **549**
abstraction, **119**
accelerators, **858**
access modifiers, **122**. *See also* access specifiers
access specifiers, **16**, 121–122
accessor methods, **144**
accumulating, **312**
accurate range checks, 265–268
`acos(x)` method, 216
action keys, **841**
`ActionEvent`s, 766, 767, 769–770, 838
  listener and handler, 840
`ActionListener` class, 766, 767, 775, 776
`actionPerformed()` method, **767**, 773–774, 881, 882, 884, 956, 957, 972
  `CardLayout` manager, 817
  `FlowLayout` manager, 811–812
actual parameters, **132**
acyclic gradients, **920**, 920–921
adapter classes, **839**
add and assign operator (+=), **312**
`add()` method, 474, **749**, 749–750, 802–803, 804, 808–809, 816, 826, 829–830
`addActionListener()` method, **767**, 775
`addItem()` method, 783, 784
`addItemListener()` method, 775
addition operator (+), 92, 93, 279
`addPoint()` method, **903**
`addSeparator()` method, 857
ad-hoc polymorphism, **559**
`AdjustmentEvent`s, listener and handler, 840
`AdjustmentListener`, 775, 776
ADTs (abstract data types), **160**
aggregation, **503**
algorithms, **445**
  bubble sort, 444–452, **445**
  insertion sort, 453–456

Allman, Eric, 18
Allman style, **18**
ambiguity, **195**, 195–196
American Standard Code for Information Interchange (ASCII), **1006**
ampersand (&), logical AND operator, **259**, 259–261, 262–265, 269, 279–280
AND operator, 259–261, 262–265, 269, 279–280
angle brackets (<>), 39
  greater than operator, 359
  less than operator, 359
anonymous classes, **227**
anonymous objects, **357**
API. *See* Application Programming Interface (API); Java API
`append()` method, **378**
applet(s), **11**, 945–982, **946**
  components, 955–961
  `destroy()` method, 962–963
  HTML documents to host, 948–950, 953, 966–968
  `init()` method, 950–955, 961–962
  life cycle, 961–968
  multimedia. *See* multimedia
  running, 947–948, 953–955
  sound, 977–980
  `start()` method, 962
  `stop()` method, 962
Applet Viewer, **947**
`appletviewer` command, **947**, 947–948
application(s)
  console. *See* console applications
  fault-tolerant, **608**
  real-time, **704**
  running, 28–29
  windowed, **11**, 11–12
application classes, 141
application files, **676**
Application Programming Interface (API). *See also* Java API
  multimedia, 968
  `Swing` containers, 832
application software, **2.** *See also* application(s)
application-triggered painting, **880**
arc(s), **896**
  drawing, 899–901, 925
architectural neutrality, **10**
`Arc2D.Float()` class, 924
`args` identifier, `String` objects, 19
argument(s), **13**, **127**
  order, 131
  passing, **13**, 19
  `print` and `println` statements, 60, 66
  `setDefaultCloseOperation()` method, 744
  superclass constructors requiring, 516–517
argument index, `printf()` method, **1016**, 1016–1017
arithmetic, floating-point, 98
arithmetic operators, shortcut, 312–317
arithmetic statements, efficient, 94
`ArithmeticException` argument, 610–611, 612
`ArithmeticException` class, 606, 619, 620, 621, 622, 627, 635, 641
array(s), 397–433, **398**, 443–488
  declaring, 398–403
  enumerations, 479–486
  initializing, 403–405
  multidimensional. *See* multidimensional arrays
  objects, 410–417
  one-dimensional (single-dimensional), **457**
  parallel, 418–421, **419**
  passed by reference, **427**
  passing to and returning from methods, 425–431
  populating, **404**
  ragged, **462**
  searching, **418**, 418–425
  `String`s, manipulating, 412
  subclass objects, 561–564
  subscripts (indexes), **399**, 399–400
  using parts, 408–409
  variable subscripts, 406–410
  wrapped, **704**
array elements, **398**
  passed by value, **426**, 426–427
  sorting using bubble sort algorithm, 444–452
  sorting using insertion sort algorithm, 453–456
`ArrayIndexOutOfBoundsException`, 635, 638
`ArrayList` class, **473**, 473–479
`Arrays` class, **465**, 465–473
ascending order, **444**
ascent, **912**

ASCII (American Standard Code for Information Interchange), **1006**
`asin(x)` method, 216
`assert` statements, **645**, 648–649
assertion(s), **645**, 645–649
`AssertionError`, 645, 648
assignment, **53**
assignment operator (=), **53**, 279
  improper use, 246
associativity, **53**
  arithmetic operators, 93
asterisk (*)
  comments, 32
  Javadoc comments, 1030
  multiplication operator, 92, 93, 279
  multiply and assign operator, **312**
  wildcard symbol, **218**
at run time, **2**
at sign (@), Javadoc tags, 1030
`atan2(x, y)` method, 216
`atan(x)` method, 216
attributes, **6**
Audio file format (.au), 977
automatic type conversion, 99–100
automatically imported constants and methods, 215–217
`AWTEvent` class, 837, 838, 843–845

B

back buffer, **828**
backslash (\)
  escape characters, 688
  escape sequences, 73
  path delimiter, 677
base classes, **503**
`BasicFileAttributes` object, 684
`BasicStroke` class, **922**
batch processing, **703**, 703–704
`BigDecimal` class, 1005
binary files, **676**
binary numbering system, **1002**, 1003, 1004
binary operators, **91**
`binarySearch()` method, 466, 468, 469
bits, **1004**
black boxes, **127**, 139
blank `final`s, **54**
blitting, **828**
block(s), **180**, 180–188
  inside (inner), **180**, 181
  outside (outer), **180**
  of code, **56**
  nested, **180**
block comments, **32**
block line transfer, **828**
Boole, George, 68, 243
`boolean` arrays, 404
Boolean operators. *See also* AND operator; NOT operator; OR operator
  precedence, 278–280
Boolean values, **243**
`boolean` variables, **67**, 67–69
border(s), drawing around a `String`, 918–919
`BorderLayout`, **753**, 753–754, 808–811, **809**, 818–821, 828–829
`BoxLayout` manager, 808, **818**
`break` keyword, 271, 272
bubble sort algorithm, 444–452, **445**
`bubbleSort()` method, 449
buffer(s), **375**, **690**
  back, **828**
  double buffering, **828**
  keyboard, **80**
  type-ahead, **80**
`BufferedInputStream` class, 691
`BufferedOutputStream` class, 691, 693, 694
`BufferedReader` class, 692, 696, 697, 700–702
`BufferedWriter` class, 692, 699, 708
`ButtonGroup` class, **781**, 781–782
`byte` array, 692
`byte` data type, **62**, 63
  type conversion, 100, 101
`ByteBuffer`s, 704–705
bytecode, **10**

C

calendar(s), Gregorian and Julian, 218
`Calendar` class, 548
call stack, **636**
  tracing exceptions, 636–641
called methods, **118**
calling
  constructors during inheritance, 514–521
  methods, **118**, 128–129
  procedures, **5**
camel casing, **53**
capacity, **473**
  `StringBuilder` objects, **375**
`capacity()` method, **376**

`CardLayout` manager, 808, **815**, 815–817, 824–826
`case` keyword, 271, 272, 273
case sensitivity, Java terms, 14
cast operators, **100**
casting, implicit and explicit, 102–104
`catch` blocks (catch clauses), **609**, 609–613, 627, 628
  multiple exceptions, 619, 620–625
catch or specify requirement, **634**
`ceil(x)` method, 216
`ChangeListener` interface, 775
changing directories, 995–996
`char` arrays, 404
`char` data type, **70**, 70–75
  type conversion, 100
character(s), **688**
`Character` class, **351**
  manipulating characters, 351–356
character values, representation, 1006–1007
`charAt()` method, 77, 352–353, **364**, **378**, 468
`checkAccess()` method, 681–682
check-digits, 153
checked exceptions, **634**
child classes, **503**
class(es), **6.** *See also specific classes*
  abstract, **548**, 548–557
  adapter, **839**
  anonymous, **227**
  application, 141
  base, **503**
  body, 142
  child, **503**
  comments, 31–34
  compiled, modifying, 29–30
  compiling, 22–28
  concept, 139–140
  concrete, **548**
  confirming storage location, 29
  containing instance fields and methods, creating, 150
  creating, 142–143
  as data types, 160–164
  declaring, 151
  defining, 14–15
  derived, **503**
  extended, **142**
  extending, 504–511
  fragile, **526**
  fundamental, **215**
  headers, 142
  identifiers, **14**, 14–16
  illegal names, 15, 16
  inner, **227**
  libraries of, **215**
  local, **227**
  method placement within, 119
  methods. *See* method(s); *specific methods*
  nested, **227**, 227–228
  nonstatic member, **227**
  objects and instantiations of, **140**
  optional, **215**
  organizing, 147–149
  parameters accepted by methods, 128
  parent, **503**
  parts, 16–17
  prewritten, importing, 217–218
  saving, 20
  `static` member, **227**
  subclasses, **503**, 508–509
  superclasses. *See* superclasses
  top-level, **227**
  type-wrapper, **88**
  unconventional names, 15, 16
  valid names, 15, 16
  virtual, **548**
class body, **17**
class clients, **141**
class definitions, **6**, 7
class diagrams, **500**, 500–503
class headers, 16
`class` keyword, 16
class methods, **208**
class users, **141**
`class` variable, 996
class variables, **209**
class-level Javadoc comments, **1030**, 1030–1031
`classpath` settings, 218
`classpath` variable, 996–997
clean builds, **30**
`clearRect()` method, **894**, 894–895
`clearRoundRect()` method, 897
client(s), class, **141**
client methods, **118**
`clone()` method, 565
`close()` method, 692, 697, 699
`closePath()` method, 925
closer in scope, **186**

closing files, **689**
collisions, **585**
colon (:), conditional operator, 276–277, 279
`Color` class, **805**, 805–808
comes into scope, **180**
command(s), **2**
command prompt, 995
comma-separated values (CSVs), **689**
comment(s), 31–34
  block, **32**
  Javadoc, **32**, **1030**, 1030–1032
  line, **32**
commenting out, **32**
`compareTo()` method, **361**, 480, 482, 686
comparison
  lexicographical, **360**
  `String` objects, 350–351, 357–361
comparison operators, **68**
compilers, **2**
  method ambiguity, **195**, 195–196
compile-time errors, **24**
compiling, 22–28, 997–998
component(s)
  `JApplet`s, 955–961
  `JPanel`s, 826–827
`Component` class, 740, 750, 805
  `FlowLayout` manager, 811
`ComponentEvent` class, 840, 844
composition, **225**, 225–227, **503**
computer files, **676**, 676–677
  checking accessibility, 681–683
  closing, **689**
  determining attributes, 684–687
  opening, **689**
  organization, 688–689
  random access (direct access; instant access). *See* random access files
  reading from, 695–697
  sequential access, **689**, 697–703
  writing to, 693–695
computer programs, **2**
  applications, **2.** *See also* application(s)
  compiling, 997–998
  development process, 3
  event-driven. *See* event-driven programs
  executing, 998
  interactive, **704**
  object-oriented, **5**, 5–6
  types, 11–12
computer simulations, **6**
concatenation, **56**, 56–58, **364**
concrete classes, **548**
conditional operator (?:), **276**, 276–277, 279
confirm dialog boxes, **89**, 89–91
console applications, **11**, 12–22
  classes, 14–17
  indent style, 17–18
  `main()` method, 18–20
  saving classes, 20
  string producing output, 13–14
`const` keyword, 15
constants, **52**
  automatically imported, 215–217
  `BorderLayout` manager, 808–809
  comparing variables, 246
  `enum`, **479**
  `final` keyword, 210–211
  literal, **52**
  named. *See* named constants
  numeric, **52**
  `PI`, 216
  prewritten, 215
  related, interfaces to store, 578–580
  scope, 56
  `StandardOpenOption` argument, 694
  unnamed, **52**
constructors, **152**, 156–159
  adding to instance methods, 281–284
  calling during inheritance, 514–521
  default, **156**, 156–157
  `Dimension` class, 912
  `Exception` class, 641–642
  `JLabel` class, 749
  `JPanel`s, 828
  `JScroll` panes, 835
  `JTextField`s, 759
  overloading. *See* overloading constructors
  with parameters, creating and calling, 196–201
  `Random` class, 1024
  superclass, requiring arguments, 516–517
consumed entries, **80**
container(s), **740**
`Container` class, 740, 804
`ContainerEvent`s, listener and handlers, 840
containment hierarchies, **802**
content pane, **802**, 802–805
controls, 740

`copyArea()` method, **903**
copying, areas of graphics, 903, 908–909
correcting syntax errors, 23–24
`cos(x)` method, 216
counter-controlled loops, **305**, 305–306
counting, **312**
crashes, **606**
`creationTime()` method, 684
CSVs (comma-separated values), **689**
curly braces ({}), 17, 21, 39
cyclic gradients, **921**

D

data compression, lossless, **969**
data fields, **142**
data files, **676**
data hiding, 153
data representation, 1001–1007
  character values, 1006–1007
  numbering systems, 1002–1004
  numeric values, 1004–1005
data types, **52.** *See also specific data types*
  abstract, **160**
  automatic promotion in method calls, 180–192
  classes as, 160–164
  enumerated, **479**
  integer, **62**, 62–67
  order of promotion, 190
  parameters accepted by methods, 128
  primitive, **52**, 160
  programmer-defined, **160**
  reference, **52**
  type conversion, **99**, 99–104
  unifying type, **99**
dead code, **134**
debugging, **3**, 31
decimal numbering system, **1002**
decimal places, specifying number to display with `printf()` method, 1015
`DecimalFormat` class, **1017**, 1017–1018
decision(s), adding to instance methods, 281–284
decision making, 241–287
  accurate range checks, 265–268
  adding decisions and constructors to instance methods, 281–284
  AND operator, 259–261, 262–265, 269, 279–280
  conditional operator, **276**, 276–277, 279
  efficient range checks, 268–269
  `if` and `if...else` structures, 244–250
  multiple statements in `if` and `if...else` clauses, 250–256
  nesting `if` and `if...else` statements, 256–259
  NOT operator, 277–278
  operator precedence, 278–280
  OR operator, 261, 262, 269, 279–280
  planning logic, 242–244
  short-circuit evaluation, 262
  `switch` statement, 270–276
decision structures, **243**
declarations, **121.** *See also* method headers
declaring
  arrays, 398–403
  classes, 151
  named constants, 54–56
  objects, 151–152
  `String` objects, 357
  variables, 53–54, 59–62
  variables in `try...catch` blocks, 616–617
decrementing variables, **305**
default constructors, **156**, 156–157
`default` keyword, 271
default packages, **584**
definite loops, **301**, 301–303, 320–321
  altering loop control variables, 305–306
`delete()` method, 683
`deleteIfExists()` method, 684
derived classes, **503**
`deriveFont()` method, 752
descending order, **444**
descent, **912**
`destroy()` method, **962**, 962–963
development environment, **10**
dialog boxes, **34**, 34–37
  confirm, **89**, 89–91
  input, **85**, 85–89
`Dimension` class, 912
direct access files, **704**
directories, **676**, 676–677
  changing, 995–996
`display()` method, 225–226, 466, 475, 521–522
divide and assign operator (/=), **312**
division
  floating-point, **92**
  integer, **92**

Division class, 605–608, 610
division operator (/), 92, 93, 279
documentation. *See* Javadoc
documentation comments, **32**
dollar sign ($), argument index, 1016
Dolphin, 994
do-nothing loops, **331**
double arrays, 404
double buffering, **828**
Double class, **372**
double data type, **69**, 69–70
  type conversion, 99, 100
double quotation mark (“), escape sequence, 73
Double.parseDouble() method, 88
double-precision floating-point numbers, **70**
do...while loops, 300, 321–324, **322**
draw objects, 884
drawArc() method, **899**, 899–900
drawImage() method, 969, 970, 975
drawing strokes, Graphics2D class, 922–923, 926–928
drawLine() method, **893**, 893–894, 906–907
drawOval() method, **898**, 898–899
drawPolygon() method, **901**, 901–902
drawRect() method, **894**, 897
drawRoundRect() method, **896**, 896–897
drawString() method, **885**, 885–893
draw3DRect() method, **897**
dual-alternative if, **247**
dummy values, **471**
dynamic method binding, 557–561, **559**
dynamic resizability, **473**

E

-ea option, 648–649
echoing the input, **78**
editing
  JTextField editability, **760**
  source code, 998
EE (Java Enterprise Edition), **994**
efficient range checks, 268–269
elements, arrays, **398**
Ellipse2D.Double() class, 924
Ellipse2D.Float() class, 924
else clause, **247**
else...if clauses, **270**
Elvis operator, 276
empty body, **304**, 304–305
empty statements, **246**
empty Strings, 361–362
encapsulation, **8**
endcap styles, **922**
endsWith() method, **364**
enhanced for loops, **407**, 412
enum, **479**, 480
enumerated data type, **479**, 479–486
equal sign (=)
  assignment operator, 53, 246, 279
  equal to operator, 68, 247, 279
  equivalency operator, **244**, 245, 246, 279, 359
  greater than or equal to operator, 68, 279
  less than or equal to operator, 68, 279
  not equal to operator, 68, 247, 279
equal to operator (==), 68, 247, 279
equals() method, **359**, 359–361, 466, 480, 565, **569**, 569–572
equalsIgnoreCase() method, **361**
equivalency operator (==), **244**, 245, 279, 359
error(s)
  class. *See* Error class; Exception class
  compile-time, **24**
  logic. *See* logic errors
  run-time, **31**
  syntax. *See* syntax errors
Error class, 604, **605**
error messages, 23, 27–28
  list, 607
escape sequences, **73**
  list, 73
event(s), **765**
  hierarchy of event classes, 837–838
  listeners associated with, 838
event handlers, **776**
  creating, 839
  list, 840
event listeners, 838
  list, 840
  Swing components, 774–777
event-controlled loops, **306**
event-driven programs, **765**, 765–774
  preparing classes to accept event messages, 766
  setEnabled() method, **770**, 770–771
  telling classes how to respond to events, 767–770
  telling classes to expect events to happen, 767
EventObject class, 837, 844
exception(s), **604**

automatically throwing, 649–650
catching, 609–619, 652–654
checked, **634**
extending classes that throw, 655–656
multiple, throwing and catching, 619–625
runtime, **604**
specification, **631**, 631–636
tracing through call stack, 636–641
unchecked, **634**
`Exception catch` blocks, 621
`Exception` class, 604, **605**, 606
constructors, 641–642
creating, 641–645, 656–661
passing on, 651–652
exception handling, 603–663, **604**
advantages, 628–631
exception specification, **631**, 631–636
`Exception` types, 644
exclamation point (!), NOT operator, **277**, 279
executing
programs, 998
statements, **2**
`exp(x)` method, 217
explicit casting, 102–104
explicit type conversion, **100**, 100–101, 102–104
extended classes, **142**
extending classes, 504–511
`extends` clause, 576
`extends` keyword, 756
eXtensible HyperText Markup Language (XHTML), **947**

F

factory methods, **678**
`false` value, 243, 404
FAQs (Frequently Asked Questions), **37**
fault-tolerant applications, **608**
fields, **688**
key, **707**
`MouseEvent` class, 848
specifying size with `printf()` method, 1015–1016
file(s). *See* computer files
file channel objects, **704**
seekability, **704**
`FileChannel` class, 704
`FileInputStream` class, 691
filenames, changing, 997
`FileOutputStream` class, 691
`Files` class, **677**, 678, 684–687, 694, 705
`FileSystem` class, 578
`fill()` method, 466
fill patterns, **920**
`fillArc()` method, **900**, 900–901
`fillOval()` method, **898**
`fillPolygon()` method, **902**, 902–903
`fillRect()` method, **894**
`fillRoundRect()` method, 897
`fill3DRect()` method, **897**
`final` constants, 54
`final` keyword
constants, 210–211
method ambiguity, 196
static and nonstatic fields, 211–215
`final` methods, inability of subclasses to override in superclasses, 530–531
`final` superclasses, inability of subclasses to override, 532
`finalize()` method, 565
`finally` blocks, **625**, 625–628
`float` arrays, 404
`Float` class, 372
`float` data type, **69**, 69–70
type conversion, 99, 101
floating-point arithmetic, 98
floating-point division, **92**
floating-point numbers, **69**, 1004
imprecision, 94–95
`floor(x)` method, 217
flow layout managers, **754**
flowcharts, **242**
`FlowLayout` class, 754–755, 808, **811**, 811–813, 822–823
`flush()` method, 692, 699
flushing, **690**
`FocusEvent`s, listener and handlers, 840
`FocusListener` class, 775, 776
folders, **676**, 676–677
font(s), 909–919
available, finding, 909–910
comparing, 915–917
drawing a border around a `String`, 918–919
font statistics, 912–914
`FontMetrics` methods, 915–919
height of, **912**
screen statistics, 912
`Font` class, **750**, 750–752, 913
`FontMetrics` methods, 915–919

for loops, 300, **317**, 317–321
  enhanced, **407**, 412
foreach loops, **408**
formal parameters, **132**
format specifiers, **1013**
format strings, **1013**
Formatter class, 1013
formatting output, 1009–1019
  DecimalFormat class, 1017–1018
  printf method, 1011–1017
  rounding numbers, 1010–1011
forward slash (/)
  comments, 32
  divide and assign operator, **312**
  division operator, 92, 93, 279
  Javadoc comments, 1030
  path delimiter, 677
fragile classes, **526**
Frequently Asked Questions (FAQs), **37**
fully quantified identifiers, **123**
functions. *See* method(s)
fundamental classes, **215**

G

garbage collector, 358
garbage value, **54**
GeneralPath class, 925
generic programming, **784**
get() method, 153, 474, 578, 972
  retrieving specific data field values, 219–220
getArray() method, 429
getAscent() method, 913
getAudioClip() method, 977
getAvailableFontFamilyNames() method, **909**, 909–911
getButton() method, 847, 850
getBytes() method, 692, 705
getClass() method, 565
getClickCount() method, 844, 847
getCodeBase() method, 969, 977
getComponent() method, 844
getContentPane() method, **802**, 802–804
getDefaultToolkit() method, **912**
getDescent() method, 913
getDocumentBase() method, 977
getFileName() method, 679
getFontMetrics() method, **913**
getGraphics() method, 885
getHeight() method, 913
getIconHeight() method, 972
getIconWidth() method, 972
getImage() method, 969
getItem() method, 781, 844
getItemAt() method, 784
getItemCount() method, 784
getKeyChar() method, 844
getLeading() method, 913
getLocalGraphics Environment() method, 910
getMaximumRowCount() method, 784
getMessage() method, 639, 642, 643
getModifiers() method, 844
getName() method, 679
getNameCount() method, 679
getPath() method, 578
getPoint() method, 844
getScreenResolution() method, **912**
getScreenSize() method, **912**
getSelectedIndex() method, 784, 785
getSelectedItem() method, 784
getSelectedObjects() method, 784
getSeparator() method, 578
getSource() method, 844
getStateChange() method, 780–781, 844
getText() method, **750**, 780
getTitle() method, 742
getWhen() method, 844
getWindow() method, 844
getX() method, 844, 847, 850
getY() method, 844, 847, 850
GIF (Graphics Interchange Format), 969
glass panes, **802**
goes out of scope, **180**
goto keyword, 15
gradient fills, **920**, 920–921
GradientPaint() method, 921
graphical user interfaces (GUIs), **6**, 801–867
  accepting input with JOptionPane class, 85–91
  applications producing GUI output, 34–37
  color, 805–808
  content pane, 802–805
  events. *See* event(s); event handlers
  JPanel class, 826–834
  JScrollPanels, **834**, 834–837
  layout managers. *See* layout managers
  menus, **851**, 851–863
  x- and y-coordinates, 845

graphics, 879–933
copying areas, 903, 908–909
creating polygons, 901–903, 925
creating shadowed rectangles, 897–898
drawing arcs, 899–901, 925
drawing lines, 893–894, 906–907, 924
drawing ovals, 898–899, 924
drawing rectangles, 894–897, 924
`drawString()` method, 885–893
fonts. *See* font(s)
`Graphics2D` class. *See* `Graphics2D` class
`paint()` methods, 880–882
`paintComponent()` method with `JPanel`s, 903–905
`repaint()` method, 880, 881, 882
`Graphics` class, **880**, 881
creating objects, 891–892
parameters, 880
screen coordinates, 892–893
Graphics Interchange Format (GIF), 969
`GraphicsEnvironment` class, 909–910
`Graphics2D` class, **920**, 920–930
drawing strokes, 922–923, 926–928
rendering attributes, 920–922
shapes, 923–925, 928–930
greater than operator (>), 68, 359
greater than or equal to operator (>=), 68, 279
Gregorian calendar, 218
`GregorianCalendar` class, 218–224, 548
Gregory XIII, Pope, 218
`GridBagLayout` manager, 808, **817**, 817–818
`GridLayout` manager, 808, **813**, 813–815, 823, 832
GUIs. *See* graphical user interfaces (GUIs)

H

hardware, **2**
has-a relationships, **225**
hash codes, **567**
hash sign (#), decimal format objects, 1017
`hashCode()` method, 565, 569
heavyweight components, **740**
height of a font, **912**
help sources, 37–38
hexadecimal numbering system, **1006**
high-level programming language, **2**
`<html>`, **949**
`</html>`, **949**
HTML (Hypertext Markup Language), **947**
hosting applets, 948–950

I

identifiers, **14**, 14–16
`if` clause, **247**
`if` statements, **244**, 244–247
equal sign, 246
nested, **256**, 256–259
relational operators, 246–247
semicolon placement, 245–246
`if...else` statements, **247**, 247–250
multiple statements in, 250–256
nested, 258–259
image(s), **969**
adding to `JApplet`s, 969–971, 975–976
`Image` class, 969, 972
`ImageIcon`s, 971–975
`ImageObserver` objects, 969
immutability, **358**
implementation, methods, **121**
implementation hiding, 8, **127**
`implements` clause, 576
`implements` keyword, 766
implicit casting, 102–104
implicit conversion, **99**
implicit type conversion, **99**, 102–104
`import` statements, **35**, 218
importing
packages, 218
prewritten classes, 217–218
`import.java.awt.Color` statement, 805
imprecision, floating-point numbers, 94–95
inclusion polymorphism, **559**
incrementing variables, **305**
indefinite loops, **301**, 306–308
indent style, 17–18
indexes, arrays, **400.** *See also* subscripts
`indexOf()` method, **363**
`IndexOutOfBoundsException`, 627
infinite loops, **301**, 301–302
information hiding, 8, **143**, 523–526, **525**
inheritance, **8**, 8–9, 499–535, **500**, 547–591
abstract classes, 548–557
accessing superclass methods, 521–523
achieving good software design, 572–573
arrays of subclass objects, 561–564
calling constructors, 514–521
diagramming using UML, 500–503

dynamic method binding, 557–561
extending classes, 504–511
information hiding, 524–526
interfaces, 574–583
methods that cannot be overridden, 526–533
multiple, **574**
`Object` class and its methods, 565–572
overriding superclass methods, 511–514
packages, 583–588
terminology, 503–504
`init()` method, 950–955, **951**, 957, 961–962, 972
initialization, **53**
arrays, 403–405
parallel arrays, 420
variables in `try...catch` blocks, 616–617
initialization lists, **404**
inlining code, **531**
inner classes, **227**
inner loops, **324**, 324–329
input dialog boxes, **85**, 85–89
`InputEvent` class, 844
`InputMismatchException`, 619, 620, 622
input/output (IO) classes, 690–697
reading from files, 695–697
writing to files, 693–695
`InputStream` class, 690, 691, 695, 700
`insert()` method, **378**
insertion sort(s), **453**
insertion sort algorithm, 453–456
inside (inner) blocks, **180**, 181
instance(s), **6**
instance methods, **144**
adding decisions and constructors, 281–284
instance variables, **142**
`instanceof` keyword, **506**, 770
instant access files, **704**
instantiation, **6**
classes, **140**
objects, 218
`int` data type, **62**, 63
returning array of, 429
type conversion, 99, 100, 101
`Integer` class, **370**, 372
integer data types, **62**, 62–67
integer division, **92**
`Integer.parseInt()` method, 88
interactive programs, **704**
interfaces, **127**, **574**, 574–583. *See also* graphical user interfaces (GUIs)
methods, 839
storing related constants, 578–580
interpreters, **2**
`intValue()` method, 371
`invalidate()` method, 612, 750
invoking methods, **118**
IO classes. *See* input/output (IO) classes
`IOException`, 627
is-a relationships, **139**, 139–140
`isAltDown()` method, 844
`isControlDown()` method, 844
`isDigit()` method, 352
`isEven()` method, 646, 647
`isLetter()` method, 352
`isLetterOrDigit()` method, 352
`isLowerCase()` method, 352
`isResizable()` method, 742
`isSelected()` method, 780, 855
`isShiftDown()` method, 844
`isUpperCase()` method, 351, 352
`isWhiteSpace()` method, 352
`ItemEvent` class, 838, 840, 844
`ItemListener` class, 775, 776
`itemStateChanged()` method, 780
iteration, loops, **300**

J

`JApplet` class, 802, **946**, 946–947
JAR (Java ARchive) files, **584**
Java, **10**, 10–12
case sensitivity of terms, 14
program types, 11–12
reserved keywords, list, 15
version names, 994
Java API, **37.** *See also* Application Programming Interface (API)
Java applications, **11**, 11–22
producing console output, analysis, 12–22
Java ARchive (JAR) files, **584**
`java` command, 29
Java Development Kit (JDK), 32, **37**, 947, **994**
Java EE Development Kit (SDK), **994**
Java Enterprise Edition (EE), **994**
Java Foundation Classes (JFCs), **740**
Java interpreter, **10**
Java Media Framework (JMF), 968
Java Micro Edition (ME), **994**

Java Platform Standard Edition 7 (Java SE 7), **994**, 994–998
Java Virtual Machine (JVM), **10**
Java Web site, 37, 38
`java.awt` package, 740, 920, 969
`java.awt.Applet` class, 977
`java.awt.Container` class, 740
`java.awt.event` package, 766, 769, 837
`javac` command, 22–23, 997
Javadoc, 1029–1036, **1030**
  comments, **32**, **1030**, 1030–1032
  generating documentation, 1032–1036
  specifying visibility of documentation, 1035–1036
Javadoc tags, **1030**
`java.lang` package, 35, 215–216, 217
`java.util` package, 218
`javax.swing` package, 215, 740
`javax.swing.JOptionPane` package, 35
`JButton` class, **760**, 760–762, 772–773
`JCheckBox` class, **778**, 778–781, 782, 786–789
`JCheckBoxMenuItem` class, 855–857
`JComboBox` class, **782**, 782–785
`JComponent`s, 740
`JDialog` class, 802
JDK (Java Development Kit), 32, **37**, 947, **994**
JFCs (Java Foundation Classes), **740**
`JFrame` class, 740, 741–748, 802
  constructing, 742
  constructors, 741–742
  customizing `JFrame` appearance, 744–745
  extending, 756–758, 764–765
  layout managers, 753–755
  methods, 742
`JFrame` component, **741**
`JFrameWithToolTip.java` file, 763
jGRASP, **10**
`JLabel` class, **748**, 748–752
`JMenu` class, 851–853
`JMenuBar` objects, 851–853
`JMenuItem` class, 852, 855–857
JMF (Java Media Framework), 968
Joint Photographic Experts Group (JPEG), 969
`JOptionPane` class, 34–37, 215
  accepting GUI input, 85–91
  `showConfirmDialog()` method, 89
`JOptionPane` component, 740
`JPanel` class, **826**, 826–834, 859–863
  components, 826–827
  constructors, 828
  `paintComponent()` method, 903–905
JPEG (Joint Photographic Experts Group), 969
`JRadioButton` class, 782
`JRadioButtonMenuItem` class, 855–857
`JScroll` panes, 834–837
  constructor, 835
`JScrollPanel`s, **834**
`JTextField`s, **758**, 758–760
  constructors, 759
  editability, **760**
Julian calendar, 218
juncture styles, **922**
JVM (Java Virtual Machine), **10**

K

K & R style, **17**
key codes, virtual, **841**
key fields, **707**
keyboard buffer, **80**
keyboard input, accepting using `Scanner` class, 76–85
`KeyEvent` class, 838, 840, 844
`KeyListener` class, 775, 776, **840**, 840–842
`keyPressed()` method, 840–842, 846
`keyReleased()` method, 840–842, 846
`keyTyped()` method, 840–842, 846
keywords, **2.** *See also specific keywords*
  reserved, list, 15

L

Landin, Peter J., 634
`lastModifiedTime()` method, 684
late method binding, 557–561, **559**
layout managers, **753**, 753–755, 808–826
  advanced, 817–818
  with fewer than five components, 821
  list, 809
leading, **912**
leaf menu items, **858**
`length` field, **407**
  two-dimensional arrays, 460–461
`length()` method, **363**
less than operator (<), 68, 359
less than or equal to operator (<=), 68, 279
lessons, downloadable, 37
lexicographical comparison, **360**
libraries of classes, **215**, 215–216

lightweight components, **740**
line(s), drawing, 893–894, 906–907, 924
line comments, **32**
`lineTo()` method, 925
`Line2D.Double()` class, 924
`Line2D.Float()` class, 924
listeners, **766**
  associated with events, 838
literal constants, **52**
literal strings, **13**
local classes, **227**
local variables, **129**
`log(x)` method, 217
logic, **2**
  decision-making, planning, 242–244
logic errors, **30**
  correcting, 30–31
logical AND operator (&&), **259**, 259–261, 262–265, 269, 279–280
logical OR operator (||), **261**, 262, 269, 279–280
`Long` class, 372
`long` data type, **62**
  type conversion, 99
look and feel, **744**
loop(s), 299–339, **300**
  `for`, 300, **317**, 317–321
  avoiding unnecessary operations, 329–330
  combining, 332
  comparing execution times for separate and fused loops, 335–336
  comparing to zero, 331–332
  counter-controlled, **305**, 305–306
  definite. *See* definite loops
  do-nothing, **331**
  `do...while`, 300, 321–324, **322**
  enhanced `for`, **407**, 412
  event-controlled, **306**
  `foreach`, **408**
  indefinite, **301**, 306–308
  infinite, **301**, 301–302
  inner, **324**, 324–329
  iterations, **300**
  nested, 324–329
  outer, **324**, 324–329
  posttest, **322**
  prefix versus postfix incrementing, 332–334
  pretest, **322**
  short-circuit evaluation, 330
  shortcut arithmetic operators, 312–317
  `while`. *See* `while` loops
loop body, **300**
  empty, 304–305
  failing to alter loop control variables within, 303–304
loop control variables, **301**
  definite loops, altering, 305–306
  failing to alter within loop body, 303–304
loop fusion, **332**
`loop()` method, 977–978
lossless data compression, **969**
low-level programming language, **2**
lvalues, **53**

M

machine code, **2**
machine language, **2**
magic numbers, **55**
`main()` method, 17, 18–20, 21–22, 118, 123, 757, 850
  application classes, 141
  calling methods, 119, 120, 131, 144
  declaring and using variables, 59–60
  `Division` class, 605, 607, 610
  lacking in applets, 950
  return types, 122
  `static` keyword, 122
`Math` class, 216–217
`Math.random()` method, 1022, 1023–1024
matrix(ces), **457**
`max(x, y)` method, 217
ME (Java Micro Edition), **994**
`MediaTracker` objects, 972
member-level Javadoc comments, **1031**, 1031–1032
memory, random access, **676**
menu(s), **851**, 851–863
  `addSeparator()` method, 857
  `JCheckBoxMenuItem` class, 855–857
  `JRadioButtonMenuItem` class, 855–857
  `setMnemonic()` method, 857–858
menu bars, **802**, 859–863
messages, methods returning, 140–141
method(s), **8**, **118**, 143–150. *See also specific methods*
  abstract, **549**
  access specifiers, 121–122
  accessor, **144**

adding parameters, 127–133
ambiguity, **195**, 195–196
`Arrays` class, 465–473
associated with objects, 140
automatically imported, 215–217
`AWTEvent` classes, 843–845
black boxes, **127**, 139
`BufferedWriter` class, 699
called, **118**
calling, 128–129
class, **208**
client, **118**
`Color` class, 806
constructors. *See* constructors
dynamic (late) method binding, 557–561, **559**
empty, 121
ending, 133
factory, **678**
`FileChannel` class, 704
implementation, **121**
instance. *See* instance methods
interfaces, 839
invoking (calling), **118**
`JCheckBox` class, 780
`JComboBox` class, 784–785
`JFrame` class, 742
`KeyListener` interface, 840–842
`Math` class, 216–217
`MouseEvent` class, 847
`MouseMotionListener` interface, 847
mutator, **144**
names, 123
nonstatic, **144**, 144–146
`OutputStream` class, 692
overloading. *See* overloading methods
overriding, **512**
parentheses, 123–124
passing a two-dimensional array to, 460
passing arrays to and returning arrays from, 425–531
`Path` class, 679
placement within a class, 119
`Random` class, 1025
receiving a single parameter, 128–130
requiring multiple parameters, 130–132
return types, 122
returning messages or values, 140–141
returning values, 133–139
signature, **131**
`static`. *See* `static` method(s)
static method binding, **559**
structure, 121
superclasses, 521–523
type, **133**, 133–134
`void`, 128
method body, **121**
method calls, **118**
automatic type promotion, 180–192
chaining, 135–136
from `println()` method, 135
virtual, **531**
method headers, **121**, 123–124
`methodGetsArray()` method, 427
.midi or .mid (Music and Instrument Interface file format), 977
`min(x, y)` method, 217
minus sign (-)
class diagrams, 501
postfix decrement operator, 313–314, **314**, 315–317, 332–334
prefix decrement operator, 313–314, **314**, 315–317, 332–334
subtract and assign operator, **312**
subtraction operator, 92, 93, 279
mission critical, **607**
mnemonics, **857**, 857–858
modules. *See* method(s)
modulus operator [mod] (%), **92**, 93, 279, 1024
`mouseClicked()` method, 846, 850
`mouseDragged()` method, 846, 847
`mouseEntered()` method, 846, 850
`MouseEvent` class, 838, 840, 844, **847**, 848
`mouseExited()` method, 846, 850
`MouseInputListener` interface, **846**
`MouseListener`, 775, 776, **846**
`MouseMotionListener` interface, 775, **846**, 847
`mouseMoved()` method, 846, 847
`mousePressed()` method, 846, 850
`mouseReleased()` method, 846, 850
`MouseWheelEvent` class, listener and handlers, 840
`moveTo()` method, 925
multidimensional arrays, 457–465, **462**
two-dimensional, 457–462, 463–465
multimedia, **968**, 968–976
adding images to `JApplet`s, 969–971, 975–976
adding sound to `JApplet`s, 977–980
`ImageIcon`s, 971–975

multiple inheritance, **574**
multiplication operator (*), 92, 93, 279
multiply and assign operator (*=), **312**
Music and Instrument Interface file format (.midi or .mid), 977
Mustang, 994
mutator methods, **144**

N

`name` variable, 77
`nameAndAddress()` method, 118–119, 122
  name, 123–124
  return types, 122
named constants, **54**
  adding to programs, 61–62
  declaring, 54–56
nested blocks, **180**
nested classes, **227**, 227–228
nested `if` statements, **256**, 256–259
nested `if...else` statements, 258–259
nested loops, 324–329
`new` keyword, 404, 558
`new` operator, **151**
`newAudioClip()` method, 977
`newByteChannel()` method, 705
`newInputStream()` method, 695
`newline()` method, 699
`newOutputStream()` method, 694
`next()` method, 77, 80
  `nextLine()` method following, 79–81
`nextBoolean()` method, 1025
`nextByte()` method, 77
`nextDouble()` method, 77, 80, 614, 1025
`nextFloat()` method, 77, 1025
`nextInt()` method, 77, 78, 79, 80, 606, 614, 1025
  wrong data type entered, 81
`nextLine()` method, 77, 615
  following another `Scanner` input method, 79–81
`nextLong()` method, 77, 1025
`nextShort()` method, 77
nonstatic fields, `final` keyword, 211–215
nonstatic member classes, **227**
nonstatic methods, **144**, 144–146, 202
nonvolatile storage, **676**
not equal to operator (!=), 68, 247, 279
NOT operator, **277**, 279
Notepad, 998
`notify()` method, 565
`notifyAll()` method, 565
null `String`s, **57**, 361–362
`null` value, 404
`NullPointerException`, 641
number(s)
  magic, **55**
  pseudorandom, **1022**
  random. *See* random numbers
  representation, 1004–1005
  rounding, 1010–1011
`NumberFormatException`, 370
numbering systems, 1002–1004
`NumbersDialog` class, 57–58
`NumbersPrintln` class, 56
numeric constants, **52**
numeric values, representation, 1004–1005

O

object(s), **6**, 6–8
  anonymous, **357**
  arrays, 410–417. *See also* array(s)
  blocks, **180**, 180–188
  concept, 139–140
  declaring and using, 151–152, 154–156
  instantiation, **6**, **140**, 218
  methods associated, 140
  properties, **407**
  reference to, **151**
  scope, 180–188
  state, **7**
`Object` class, **565**, 565–572, 679, 690
object references, 563–564
`object` tag attributes, 949
object-oriented program(s), **5**, 5–6
object-oriented programming (OOP), 5–9
  classes, **6**
  encapsulation, **8**
  inheritance, **8**, 8–9
  objects, **6**, 6–8
  polymorphism, **9**
  procedural programming compared, 9
octothorpe, **1017**
one-dimensional arrays, **457**
OOP. *See* object-oriented programming (OOP)
`open()` method, 704
opening files, **689**
operands, **91**
operator precedence, **93**, 278–280

optional classes, **215**
OR operator, 261, 262, 269, 279–280
`ordinal()` method, 480
`out` object, 14
out of bounds subscripts, **400**
outer loops, **324**, 324–329
`OutputStream` class, 690, 691, 692, 693, 694
outside (outer) blocks, **180**
ovals, drawing, 898–899, 924
overloading constructors, 157, 197–198, 199–201
  efficiency, `this` reference, 205–206, 207–208
overloading methods, 132, **188**, 188–194
  automatic type promotion, 180–192
overriding
  `final` superclass methods, 530–531
  methods, **512**
  superclass methods, 511–514
  variables, **183**, 183–184

P

package(s), **35**, **215**, 215–216, 583–588. *See also specific packages*
  default, **584**
  importing, 218
`package` access specifier, 121
`paint()` method, **880**, 880–882, 895, 920, 961, 969, 970, 974, 975
`paintComponent()` method, 903–905
`paintIcon()` method, 972
painting, **880**
parallel arrays, 418–425, **419**
  initializing, 420
parameters, **127**
  actual, **132**
  adding to methods, 127–133
  formal, **132**
  `Graphics` objects, 880
  superclasses as method parameter types, 559–560
parent classes, **503**
parentheses (()), 39
  method headers, 123–124
`parseDouble()` method, 372, **372**, 701
`parseFloat()` method, 372
`parseInt()` method, 220, **370**, 370–371, 372, 701, 710, 716
`parseLong()` method, 372
parsing, **27**, **88**
Pascal casing, **15**
passed by reference, **427**
passed by value, **426**, 426–427
passing arguments, **13**
path(s), **677**
  absolute, **679**, 680–681
  relative, **679**, 680–681
`Path` class, **677**, 677–681, 695
  creating objects, 678–679
  relative, converting to absolute, 680–681
  retrieving information about, 679–680
path delimiters, **677**
pattern `String`, **1017**
percent sign (%)
  format specifiers, 1013
  remainder and assign operator, **312**
  remainder (modulus) operator, **92**, 93, 279, 1024
permanent storage devices, **676**
`PI` constant, 216
pipe (|), logical OR operator, **261**, 262, 269, 279–280
pixels, **743**
`play()` method, 977
plus sign (+)
  add and assign operator, **312**
  addition operator, 92, 93, 279
  class diagrams, 501
  concatenation, 57, 365
  postfix increment operator, **312**, 313–314, **314**, 315–317, 332–334
  prefix increment operator, **312**, 313–314, **314**, 315–317, 332–334
PNG (Portable Network Graphic), 969
point size argument, **751**
`Point2D.Double()` class, 924
`Point2D.Float()` class, 924
polygons, creating, 901–903, 925
polymorphism, **9**, 57, **511**, 551
  ad-hoc, **559**
  pure (inclusion), **559**
  subtype, **512**
populating an array, **404**
Portable Network Graphic (PNG), 969
`position()` method, 704
postfix decrement operator (postfix --), 313–314, **314**, 315–317, 332–334
postfix increment operator (postfix ++), **312**, 313–314, **314**, 315–317, 332–334
posttest loops, **322**

pound sign (#), decimal format objects, 1017
`pow(x, y)` method, 217
precedence
  arithmetic operators, 93, 278–280
  Boolean operators, 278–280
preferred size, `Component`s, **811**
prefix decrement operator (prefix --), 313–314, **314**, 315–317, 332–334
prefix increment operator (prefix ++), **312**, 313–314, **314**, 315–317, 332–334
pretest loops, **322**
prewritten classes, importing, 217–218
primary key, **147**
primary surface, **828**
priming input, **309**
priming reads, **309**
primitive data types, **52**, 160
  converting to `String`s, 364
`print()` method, 14, 365
  arguments, 60
  displaying variables or constants, 56
  literal `String`s contained in, 78
`printf()` function, C programming language, 1013
`printf()` method, 1011–1017
  optional argument index, **1016**, 1016–1017
  specifying field size, 1015–1016
  specifying number of decimal places to display, 1015
`println()` method, 14, 17, 28, 29, 73–74, 118, 119, 365, 1012
  arguments, 60, 66
  displaying characters, 71–72
  displaying variables or constants, 56
  method calls from, 135
`printStackTrace()` method, 639, 640
`PrintStream` class, 14, 66, 691, 692
private access, **143**
`private` access specifier, 121, 143
`private` classes, 228
`private` keyword, 524–525
`private` methods, 145, 146
`private protected` access specifier, 143
procedural programming, **5**
  object-oriented programming compared, 9
procedures, **5.** *See also* method(s)
program(s). *See* computer programs
program comments, **31**, 31–34
program files, **676**
program statements, **2**
programmer-defined data types, **160**
programming
  object-oriented. *See* object-oriented programming (OOP)
  procedural, **5**
promotions, **99**
prompt, **78**
properties, **6**
  objects, **407**
`protected` access specifier, 121
`protected` keyword, **525**
pseudocode, **242**
pseudorandom numbers, **1022**
`public` access specifier, 121, 122
`public` keyword, 16, 19, 525
`public` methods, 145, 146
pure polymorphism, **559**

Q

question mark (?), conditional operator, 276–277, 279

R

ragged arrays, **462**
RAM (random access memory), **676**
random access files, 703–707, **704**, 714–729
  accessing randomly, 715–717, 728–729
  displaying statistics, 725–726
  multiple, creating, 718–720
  reading records sequentially, 714–715, 726–728
  writing records to, 707–713
random access memory (RAM), **676**
`Random` class, 1022, 1024–1027
`random()` method, 217, 1023
random numbers, 1021–1027, **1022**
  generated by computers, 1022
  `Math.random()` method, 1022, 1023–1024
  `Random` class, 1022, 1024–1027
range checks, **265**
  accurate, 265–268
  efficient, 268–269
range matches, **422**
  searching arrays, 421–423
`read()` method, 695, 697, 704
`readAttributes()` method, 684
`Reader` class, 690, 692

ReadFile class, 695, 696
readLine() method, 696, 697, 699
real-time applications, **704**
records, **688**, 688–689
rectangles
  drawing, 894–897, 924
  shadowed, creating, 897–898
Rectangle2D.Double() class, 924
Rectangle2D.Float() class, 924
redeclaring variables, **182**
reference(s), **202**, **350.** *See also* methods
  arrays passed by, **427**
  to the object, **151**
reference types, **52**, 151, **427**
  list, 52
regionMatches() method, **367**
relational operators, **68**
  if statements, 246–427
relative paths, **679**
  converting to absolute paths, 680–681
remainder and assign operator (%=), **312**
remainder (modulus) operator (%), **92**, 93, 279, 1024
remove() method, 474, **750**, 803
removeAllItems(), 784
removeItem(), 784
rename command, 997
rendering attributes, Graphics2D class, 920–922
repaint() method, 750, **880**, 881, 882
replace() method, **364**
rerendering, **880**
reserved keywords, list, 15
return clauses, multiple, 135
return statements, **133**, 134–135, 429
  void methods, 134
return types, **122**
return types, 635
returning values, **122**
rewind() method, 705
rint(x) method, 217
robustness, **608**
root directory, **676**
round(x) method, 217
rounding numbers, 1010–1011
running applications, 28–29
run-time errors, **31**
runtime exceptions, **604**
RuntimeExceptions, 635
rvalues, **53**

S

sandboxes, **949**
saving
  classes, 20
  source code, 998
Scanner class, accepting keyboard input, 76–85
scientific notation, **69**
scope, **56**, 180–188
  variables closer in, **186**
  variables coming into, **180**
  variables going out of, **180**
screen statistics, 912
SDK (Java EE Development Kit), **994**
searching arrays, **418**, 418–425
  for range matches, 421–423
seeds, **1022**
seekability, file channels, **704**
semantic errors, **4**
semicolons (;), if statements, 245–246
sequence structures, **242**
sequential access files, **689**, 697–703
set classpath= command, 997
set() method, ArrayList class, 474
set methods, 153
setAlignment() method, 811–812
setBackground() method, 804, 805
setBounds() method, 742, 757
setCharAt() method, **378**
setColor() method, 887, 890, 920
setDefaultCloseOperation() method, 744, 756
setDefaultLookAndFeelDecorated() method, 744–745
setEditable() method, **760**, 784, 785
setEnabled() method, **770**, 770–771
setFont() method, **750**, 886–887, 890
setForeground() method, 805
setJMenuBar() method, 853
setLayout() method, 804, 811, 956
setLayoutManager() statement, 803
setLength() method, **375**, 377
setLocation() method, 757, **882**, 882–884
setMaximumRowCount() method, 784
setMnemonic() method, 857–858
setPaint() method, 921
setResizable() method, 742
setSelected() method, 780, 855
setSelectedIndex() method, 784
setSelectedItem() method, 784

setSize() method, 742, 756
setStroke() method, **922**, 922–923
setText() method, **750**, 780, 956
setTitle() method, 742, 756
setToolTipText() method, **762**
setVisible() method, 742, 743–744, 756–757
shadowed rectangles, creating, 897–898
shadowing, **184**, 184–185
shapes, Graphics2D class, 923–925, 928–930
short data type, **62**, 63
  type conversion, 100
short-circuit evaluation, **262**
  order, 330
show() method, 744
showConfirmDialog() method, **89**
showInputDialog() method, **85**, 85–88, 370
showMessageDialog() method, 14, 35, 57
signatures, methods, **131**
significant digits, **69**
simulations, **6**
sin(x) method, 217
single quotation mark ('), escape sequence, 73
single-alternative if, **247**
single-dimensional arrays, **457**
single-precision floating-point numbers, **70**
size() method, 474, 684, 704
skip() method, 697
sleep method, 319
software, **2.** *See also* computer programs
software development, inheritance to achieve, 572–573
software development kits (SDKs), **37**, 994
sort() method, 465, 466, 469
sorting, **444**
  bubble sort algorithm, 444–452, **445**
  insertion sort algorithm, **453**, 453–456
sound, adding to applets, 977–980
source, events, **766**
source code, **10**
  Notepad, 998
  TextPad, 998–999
speak() method, 549, 550–551, 558–559, 561–562, 575–577
split() method, 701
sqrt(x) method, 217
square brackets ([]), 39
stack backtrace, 608
stack trace (stack trace history list), **607**
stack traceback, 608
standard arithmetic operators, **91**, 91–94, 96–98
  associativity and precedence, 93–94, 278–280
standard input devices, **76**
standard output devices, **14**
StandardOpenOption argument, 694
start() method, 957, 961, **962**
startsWith() method, **364**
state, **7**
statements, 13–14. *See also specific statements*
  commenting out, **32**
  empty, **246**
  unreachable, **134**
static access specifier, 143
static fields, 202, 208–215
  constants, 210–211
  final keyword, 211–215
static import feature, **682**
static keyword, **19**, 88, 122
  data fields, 142–143
static member classes, **227**
static method(s), 88, 144–145, 146, 466
  accepting arguments and returning values, 137–138
  inability of subclasses to override in superclasses, 526–530
  requiring no arguments and returning no values, 124–126
static method binding, **559**
stop() method, **962**, 977
storage
  nonvolatile, **676**
  volatile, **676**
streams, **689**, 689–690
String argument, 360
String class, 19, **72**, **351**, 352–353, 357–370, 700, 701, 914
  args identifier, 19
  comparing, 350–351, 357–361
  comparing Strings, 247
  concatenating with numeric values, 56–58
  converting to double values, 372
  converting to numbers, 370–374
  declaring, 357
  empty and null Strings, 361–362
  immutability, **358**
  manipulating arrays, 412
  null elements, 404
String input, 83–85
String variables, **357**

StringBuffer class, 351, **374**, 374–375
StringBuilder class, 351, **374**, 374–381
stringWidth() method, **914**
strokes, **922**
strongly typed language, **53**
stubs, **121**
style argument, **751**
subclasses, **503**, 508–509
  arrays of subclass objects, 561–564
  inability to override static methods in superclasses, 526–530
subroutines. *See* method(s)
subscripts
  arrays, **399**, 399–400
  out of bounds, **400**
  variable, 406–410
substring() method, **365**
subtract and assign operator (-=), **312**
subtraction operator (-), 92, 93, 279
subtype polymorphism, **512**
Sun Microsystems Audio file format (.au), 977
super keyword, **517**
  this keyword compared, 523
super() method, 517, 756
superclasses, **503**, 507–508
  constructors requiring arguments, 516–517
  as method parameter types, 559–560
  methods, 521–523
  overriding methods, 511–514
  that cannot be overridden, 526–533
Swing class, 740
Swing components, 739–793, **740**
  adding JButtons, 760–762
  adding JTextFields, 758–760
  associated listener-registering methods, 776
  ButtonGroup class, 781–782
  event listeners, 774–777
  event-driven programming, 774–777
  extending JFrame class, 756–758, 764–765
  JCheckBox class, 778–781, 786–789
  JComboBox class, 782–785
  JFrame class, 741–748
  JLabel class, 748–752
  layout managers, **753**, 753–755
  tool tips, 762–763
Swing containers, Application Programming Interface, 832
switch keyword, 271
switch statements, 270–276, **271**
symbolic constants, **54.** *See also* named constants
syntactic salt, **634**
syntactic sugar, **634**
syntax, **2**
syntax errors, **3**
  correcting, 23–24
  error messages, 23, 27–28
System class, 14, 35, 141, 215, 692
system software, **2**
System.err object, 692
System.exit() statements, 34, 626, 628
System.getProperty() method, 699, 709
System.in object, 76, 695
System.out object, 76, 692, 693
System.out.format() method, 1013
System.out.printf() method, 1011–1017, **1012**
system-triggered painting, **880**

T

tables, **457**
tag(s)
  HTML, **949**, 949–950
  Javadoc, **1030**
tag attributes, **949**
tan(x) method, 217
temporary variables, 676
ternary operators, **276**
text files, **676**
TextEvents, 838
  listener and handler, 840
TextPad, 998–999
this keyword, super keyword compared, 523
this() method, 201–208, **202**
  calling, 206
  overloaded constructor efficiency, 205–206, 207–208
threads of execution, **375**
throw statements, **609**, 609–610
Throwable class, 639, 641
ThrowableException, 612
throws clause, 634, 635
Tiger, 994
toAbsolutePath() method, 680–681
TOCTTOU bug, **683**
tokens, **76**
toLowerCase() method, 352, **363**
tool tips, **762**, 762–763
Toolkit class, 912

ToolTipDemo.java file, 763
top-level classes, **227**
top-level containers, **802**
`toString()` method, **364**, 365, 478, 480, 565, **566**, 566–569, 679
`toUpperCase()` method, 351, 352, **363**
`true` value, 243
`try` blocks, 609, 614–615, 627, 628, 639, 692
  multiple exceptions, 619–621
`try...catch` blocks, declaring and initializing variables, 616–617
`try...catch` code, 613
`try...catch` sequences, 610
  end, 626–628
two-dimensional arrays, **457**, 457–462, 463–465
  `length` field, 460–461
  passing to a method, 460
  ragged, 462
type casting, **100**
type conversion, **99**, 99–104
  automatic, 99–100
  explicit, **100**, 100–101, 102–104
  implicit, **99**, 102–104
type-ahead buffer, **80**
typeface argument, **750**, 750–751
type-safe values, **483**
type-wrapper classes, **88**

U

UI components, **740.** *See also* `Swing` components
UML. *See* Unified Modeling Language (UML)
unary cast operators, **100**
unary operators, **100**
unchecked exceptions, **634**
Unicode, **15**, **1006**
  values, list, 71–72
Unified Modeling Language (UML), **500**
  diagramming inheritance, 500–503
unifying type, **99**
uninitialized variables, **54**, 61
unique identifiers, 148
Universal Resource Locators (URLs), 969
unnamed constants, **52**
unreachable statements, **134**
upcasting, **506**
upper camel casing, **15**
URLs (Universal Resource Locators), 969
users, class, **141**

V

`validate()` method, 612, 750, 957
validating data, **308**
  loops, 308–311
value(s)
  array elements passed by, **426**, 426–427
  Boolean, **243**
  character, representation, 1006–1007
  comma-separated, **689**
  `denominator`, 613
  dummy, **471**
  `false`, 243, 404
  garbage, **54**
  lvalues, **53**
  methods returning, 133–139, 140–141
  `null`, 404
  numeric, representation, 1004–1005
  passed by, 426–427
  returning, **122**
  rvalues, **53**
  `true`, 243
  type-safe, **483**
  Unicode, list, 71–72
`valueOf()` method, 371, 372, 480, 482
`values()` method, 480
  variable(s), **5**, **52.** *See also specific variables*
  blocks. *See* block(s)
  `boolean`, **67**, 67–69
  class, **209**
  closer in scope, **186**
  comparing to constants, 246
  declaring, 53–54, 59–62
  declaring and initializing in `try...catch` blocks, 616–617
  decrementing, **305**
  holding single value at a time, 58
  incrementing, **305**
  instance, **142**
  local, **129**
  overriding, **183**, 183–184
  redeclaring, **182**
  scope. *See* scope
  shadowing, **184**, 184–185
  temporary, 676
  uninitialized, **54**, 61
  using names multiple times, 183
variable declarations, **53**
viewports, **834**
virtual classes, **548**

virtual key codes, **841**
virtual method calls, **531**
`void` keyword, **19**
`void` methods, 128, 134
`void` return type, 122
volatile storage, **676**

W

.wav (Windows Wave file format), 977
Web browsers, **948**
`while` loops, 300, **301**, 301–311
  altering loop control variable, 303–304, 305–306
  definite, 301–303
  empty body, **304**, 304–305
  indefinite, 306–308
  validating data, 308–311
whitespace, **17**
widgets, 740
wildcard symbol (*), **218**
`Window` class, 740–741
window decorations, **744**
windowed applications, **11**, 11–12
`WindowEvent` class, 838, 840, 844
`WindowListener` interface, 775
Windows Wave file format (.wav), 977
WORA ("write once, run anywhere"), **10**
`work()` method, 575–577
`wrap()` method, 704
wrapped arrays, **704**
wrappers, **370**
`write()` method, 692, 699, 704, 708
"write once, run anywhere" (WORA), **10**

X

x-axis, **845**
x-coordinate, **845**
XHTML (eXtensible HyperText Markup Language), **947**

Y

y-axis, **845**
y-coordinate, **845**

Z

zero, comparing to, 331–332