

---

# INDEX

---

## A

- abs () method, 199–201
- Abscissas, 322
- AbsDemo class, 200–201
- Absolute value methods, 199–201
- Abstract classes, 87–89
  - compared to interfaces, 132
  - using, 88–89
- Abstract methods, 123–24
- Abstract Window Toolkit (AWT), 4, 176
- AbstractGas class, 227–28, 229
- Access control, and packages, 144
- Access modifiers:
  - and inheritance, 84–85
  - methods, 114–16
  - variables, 102–3
- Access privileges, 67–68
  - types of access, 67
- AccessDemo class, 115–16
- AccessDriver class, 116
- Accessibility API, 4
- acos () method, 204
- ActionListener class, 78
- ActionListener interface, 428–29
- Air class, 160–61, 233–34
- Air2 class, 165
- Air3 class, 166–67
- AirDriver class, 233–34
- Aliasing, 346–47
- Angle of attack, 327
- Angles, converting, 207
- AnonDemo class, 80–81
- Anonymous inner classes, 77, 80–81
- Apache server, 438
- ArgDemo class, 137–38, 168–69
- Arguments, passing by value, 16
- Arithmetic operators, 44–45
- ArithOpDemo class, 45
- Array elements:
  - accessing, 166–67
  - initializing, 164–67
- ArrayList class, 170
- Arrays, 159–72
  - accessing an element in, 17
  - array elements:
    - accessing, 166–67
    - initializing, 164–67
  - collection classes in the Java API, 170–71
  - defined, 159
  - indices, 16–17
  - length, 169–70
  - as method arguments and return types, 167–69
  - of more than two dimensions, 164
  - moving from C to Java, 26–27
  - moving from Fortran to Java, 16–17
  - one-dimensional, 160–61
  - passing as a method argument, 168–69
  - size of, 16
  - two-dimensional, 161–64
- asin () method, 204
- Assignment operators, 45–46
- AssignOpDemo class, 46
- atan () method, 204
- AtmGUI class, 421–23
  - final form of, 430–33
- AtmServlet class, 443–47
  - source code, 444–47
- AudioInputStream class, 389
- available () method, InputStream class, 385

## B

- Basic syntax, *See* Syntax:
- BigDecimal class, 197
- BigInteger class, 197
- Binary operators, 44

- Bins, 347–49
  - Bitwise operators, 50–51
  - BitWiseDemo class, 51
  - BlackBody class, 96–97, 119
  - BlackBody2 class, 104
  - BlackBody2Driver class, 104–5
  - BlackBody3 class, 105–6
  - BlackBody4 class, 128–29
  - Blottner curve fit expression, 203
  - Boolean class, 185
  - boolean data type, 14–15, 24, 93
  - Boolean operators, 48–50
  - BooleanOpDemo class, 50
  - BorderLayout class, 426
  - Braces, 13
  - break statements, 60–61
  - BreakDemo class, 61
  - Brigham, E. Oran, 334
  - BufferedInputStream class, 384, 385–86
  - BufferedOutputStream class, 390, 391
  - BufferedReader class, 175, 397–98
  - BufferedWriter class, 175, 400–401
  - Built-in math functions, 19, 197–214, *See also* User-defined math functions
    - absolute value methods, 199–201
    - angles, converting, 207
    - comparing built-in math capability of C/C++/Fortran/Java, 212–14
    - conversion methods, 206–7
    - java.math package, 212
    - Math class, 198
    - mathematical constants, 199
    - maximum methods, 208–9
    - minimum methods, 208–9
    - moving from C to Java, 29
    - moving from C++ to Java, built-in math functions, 34–35
    - pow() method, 201–2
    - random number generator methods, 211–12
    - remainder methods, 209–11
    - rounding methods, 209–11
    - sqrt() method, 201–2
    - StrictMath class, 198
    - transcendental math functions, 203–4
    - trigonometric methods, 204–6
  - Byte class, 184–86
  - byte data type, 14–15, 24, 93
  - Byte input streams:
    - AudioInputStream class, 389
    - BufferedInputStream class, 384, 385–86
    - ByteArrayInputStream class, 384, 386
    - DataInputStream class, 384, 386–87
    - FileInputStream class, 384, 387–88
    - FilterInputStream class, 384, 388
    - InputStream class, 385
    - ObjectInputStream class, 384, 388–89
    - PipedInputStream class, 389
    - SequenceInputStream class, 389
    - StringBufferInputStream class, 389
  - Byte output streams, 389–96
    - BufferedOutputStream class, 391
    - ByteArrayOutputStream class, 391–92
    - DataOutputStream class, 392–93
    - FileOutputStream class, 393
    - FilterOutputStream class, 393
    - ObjectOutputStream class, 393–95
    - OutputStream class, 390–91
    - PrintOutputStream class, 395–96
  - Byte streams, 20, 382
  - ByteArrayInputStream class, 384, 386
  - ByteArrayOutputStream class, 391–92
  - Bytecode, 6
- C**
- C/C++, 442
  - Camber, 327
  - CamberFunction class, 329
    - constructor, 329
  - CardLayout class, 426
  - CastDemo class, 107–8
  - Casting, 106–8
  - catch clause, 152–53
  - Catenary, 222–23
  - ceil() method, 209
  - ChainDemo class, 125
  - char data type, 14–15, 24, 93
    - double data type, 24
  - Character class, 185
  - Character input streams, 396–99
    - BufferedReader class, 397–98
    - CharArrayReader class, 399
    - FileReader class, 398
    - FilterReader class, 399
    - InputStreamReader class, 383, 398–99
    - PipedReader class, 399
    - PushbackReader class, 399
    - Reader class, 396–97
    - StringReader class, 399
  - Character output streams, 399–403
    - BufferedWriter class, 400–401
    - CharArrayWriter class, 403
    - FileWriter class, 401
    - FilterWriter class, 403
    - OutputStreamWriter class, 401–2
    - PipedWriter class, 403
    - PrinterWriter class, 402–3
    - StringWriter class, 403
    - Writer class, 399–400

- Character streams, 20, 382
- CharArrayReader class, 399
- CharArrayWriter class, 403
- Chord, 327
- Chord line, 327
- CircCylGrid class, 162–63
- .class files, using the CLASSPATH to locate, 143–44
- Class hierarchies, 225–46
  - AbstractGas class, 227–28, 229
  - Air class, 233–34
  - gas mixture, defining state/behavior of, 226–27
  - general class hierarchy structure, 227–29
  - N class, 241–42
  - N2 class, 240–41
  - NitrogenGas class, 242–45
  - PerfectGas class, 227–28, 230–33
  - RealGas class, 227–28, 234–38
  - Species class, 238–40
    - class hierarchy, 228
- Class loader, 7
- Class method, 70
- Class variables, 24, 69–70, 91, 95–98
- Classes, 12–13, 22, 23, 37, 65–90, *See also* Abstract classes; Anonymous inner classes; Final classes; Instance (inner) nested classes; Nested classes; Static nested classes
  - abstract, 87–89
    - using, 88–89
  - access privileges, 67–68
  - constructors, 65, 72–74
  - declaration syntax, 66–67
  - defined, 65
  - encapsulation, 81–82
  - fields, declaring, 69–70
  - final, 89
  - garbage collector, 89–90
  - inheritance, 82–84
  - initializers, 65
  - members, 65
  - methods, declaring, 70–71
  - nested, 77–81
  - and object structure, 13
  - objects, 68–69
    - copying, 75–77
  - static initialization blocks, 74
  - super keyword, 86
  - this keyword, 86–87
- CLASSPATH environment variable, 143–44
- Client machine, 436
  - sending output back to, 443
- Client-server relationship, 437
- close() method:
  - InputStream class, 385
  - OutputStream class, 391
  - Reader class, 396–97
  - Writer class, 400
- CmdLineDemo class, 408–9
- Code blocks, 13
- Code libraries, moving from Fortran to Java, 18–19
- Collection classes, 170–71
- Collection interface, 137
- Collision integrals, 373
- Color class, 176
- Command line arguments, getting input, 407–9
- CommandLine class, 120
- Comment statements, 13
- CommentDemo class, 43
- Common Gateway Interface (CGI), 7
- Comparator, 191
- Comparator, defined, 191
- compareTo() method, 190
- compareToIgnoreCase() method, 190
- Compiler, 6, 8–9
  - and exception handling, 18
  - obtaining, 7
  - options, 8
  - output of, 9
- Compressible boundary layer (example problem), 296–305
  - CompressODE class, 300–302
  - ShootingCompress class, 303–5
- CompressODE class, 300–302
- computeGaussWeights() method, 323–24
- computeMolarMass() method, 235
- computePressure() method, 404
- computeTemperature() method, 404
- ConcatDemo class, 189–90
- Constants, 133
- Constructors, 72–74
  - classes, 65
- Content panes, 420–21
- continue statements, 61–62
- Control statements, 54–60
  - transfer of, 60–62
    - break statements, 60–61
    - continue statements, 61–62
    - return statements, 62
- ConvertDemo class, 207
- CORBA Object Request Broker (ORB), 4
- Core J2SE libraries, 174–76, *See also* Java 2 Platform Standard Edition (J2SE)
  - java.io package, 174–75
  - java.lang package, 174–75
  - java.math package, 174–75
  - java.util package, 175–76
- Core JDBC API, 174
- cos() method, 204
- CreateDemo class, 69

- CreateString class, 188–89
- CreateWrapper class, 183–84
- Crococo's method, 304
- Curve fitting specified heat data (example problem), 370–73
- D**
- Damped spring motion (example problem), 280–85  
 solution, 283–85  
 SpringODE class, 282–83
- Damping forces, 281
- Data modeling, 365–79  
 curve fitting specified heat data (example problem), 370–73  
 DataModeling class, 368–69  
 least squares fit to a polynomial equation, 366–68  
 nonpolynomial equations, fitting to, 373–74  
 Polynomial class, 369–70  
 Power class, 374–78
- Data streams, 382–83
- Data types, moving from C++ to, 33
- DataInputStream class, 384, 386–87
- DataModeling class, 354, 358, 368–69, 374
- DataOutputStream class, 392–93
- Decimation in Frequency (DIF) technique, 349
- Decimation in Time (DIT) technique, 349
- DeclareDemo class, 99–100, 113–14
- Default access, 67–68  
 variables, 102
- Deitel, Harvey M., 124
- Deitel, Paul J., 124
- Differential equations, 271–307  
 compressible boundary layer (example problem), 296–305  
 damped spring motion (example problem), 280–85  
 solution, 283–85  
 SpringODE class, 282–83  
 defined, 271  
 embedded Runge-Kutta algorithms, 285–91  
 initial value problems, 275–76  
 ODE class, 273–75  
 ODESolver class, 279–80  
 ordinary, 272–73  
 first-order ODE, 272–73  
 second-order ODE, 272  
 partial, 306  
 Runge-Kutta schemes, 276–80  
 rungeKutta4() method, 279–83  
 shooting methods, 292–96  
 two-point boundary problems, 292, 305–6
- Dimension class, 176
- Discrete Fourier transform (DFT), 333–34, 336–43  
 defined, 336  
 discreteFT() method, 338–41  
 TestDF class, 340–41  
 discreteFT() method, 338–41, 343, 354  
 do-while loops, 56–57  
 do-while statement, 14  
 Documentation comments, 13  
 Double class, 184–86  
 double data type, 14–15, 16, 24, 93, 208  
 doubleValue() method, 183–84, 185  
 DoWhileDemo class, 57  
 Dynamic memory allocation, 17  
 moving from C to Java, 27–28
- E**
- Eckel, Bruce, 225
- Electrostatic interface, 133
- Embedded Runge-Kutta algorithms, 285–91  
 embeddedRK5() method, 287–91  
 EmbedSpring class, 290–91  
 Empty statement, 112  
 Encapsulation, 37–38, 81–82, 111  
 Enterprise JavaBeans, 30  
 Enthalpy ratio, 299  
 Enumerations, moving from C++ to Java, 34  
 EqnSolver class, 249  
 methods, testing, 263–65  
 equals() method, 191  
*Essential JNI: Java Native Interface* (Gordon), 124  
 Euler's number, 199  
 Event handlers, 427–30  
 EventDemo class, 79–80  
 Exception handling, 149–57  
 Exception class hierarchy, 150–51  
 moving from C to Java, 28  
 moving from Fortran to Java, 18  
 passing exceptions to another method, 156–57  
 throw and throws keywords, 155–57  
 try statements, 151–55  
 catch clause, 152–53  
 finally clause, 153  
 try keyword, 152  
 using, 154–55  
 exp() method, 203–4  
 Extensible Markup Language (XML), 173–78  
 Java API for, 4
- F**
- Fast Fourier Transform and Its Applications* (Brigham), 334  
 Fast Fourier transforms (FFTs), 334, 349–54  
 Decimation in Frequency (DIF) technique, 349  
 Decimation in Time (DIT) technique, 349

- defined, 349–50
  - `fastFFT()` method, 351–53
  - implementing as a Java method, 351
  - `TestFFT` class, 353–54
  - `fastFt()` method, 351–53
  - `fastFt()` method, 351–54
  - `fastFFT()` method, 354
  - Fields, 12–13
    - declaring, 69–70
  - `FileDemo` class, 412–14
  - `FileInputStream` class, 384, 387–88
  - `FileOutputStream` class, 393
  - `FileReader` class, 398
  - `FileWriter` class, 399, 401
  - `FilterInputStream` class, 384, 388
  - `FilterOutputStream` class, 390, 393
  - `FilterReader` class, 399
  - `FilterWriter` class, 403
  - Final classes, 89
  - final keyword, 89
  - Final methods, 124
  - Final, methods, 124
  - Final variables, 105–6
  - finally clause, 153
  - First-order ODE, 272–73
  - `FithOrderDemo` class, 371–73
  - Float class, 184–86
  - float data type, 14–15, 24, 93, 208
  - `floor()` method, 209
  - FlowLayout layout manager, 426
  - `Fluids.Gas` package, 141–42
  - `flush()` method:
    - `OutputStream` class, 391
    - `Writer` class, 400
  - Font class, 176
  - for loops, 57–58
  - `ForDemo` class, 58
  - Fortran, 442
  - Forward transforms, 335–36
  - Fourier class, 339–40
  - Fourier transforms, 333–54
    - composite signals, analyzing, 343–45
    - defined, 333
    - discrete Fourier transform (DFT), 333–34, 336–43
      - defined, 336
      - `discreteFT()` method, 338–41, 354
      - `TestDF` class, 340–41
    - fast Fourier transforms (FFTs), 334, 349–54
      - Decimation in Frequency (DIF) technique, 349
      - Decimation in Time (DIT) technique, 349
      - defined, 349–50
      - `fastFFT()` method, 351–53, 354
      - implementing as a Java method, 351
      - `TestFFT` class, 353–54
    - forward, 335–36
    - generic part implementation, 354
      - `leastSquaresFit()` method, 354, 361, 363
    - inverse, 333, 335–36
    - sampling theory, 345–47
      - aliasing, 346–47
      - signal characterization, 334–35
      - spectral leakage, 347–49
    - time and frequency function relationships, 336
  - Fredholm integrals, 308, 326
  - Free-form coding, 13
  - Free variables, 273
  - Freely Distributable Math Library (`fdlibm`), 19
  - Full pivoting, 250–51
  - Function pointer, 355
  - Functions:
    - integration of, 307–31
    - moving from C to Java, 25–26
    - moving from Fortran to Java, 15–16
- ## G
- Gamma function, 219–20
  - Garbage collector, 17, 89–90
  - `GasDriver` class, 71
  - `GasDriver3` class, 73–74
  - `GasDriver4` class, 76–77
  - `GasDriver5` class, 95
  - Gauss-Jordan elimination, 253–55, 261
  - Gauss-Legendre formula, 323
  - Gaussian elimination, 255–57
  - `gaussian()` method, 256–57, 261, 267
  - Gaussian quadrature methods, 322–26
  - `gaussJordan()` method, 254–56
  - `gaussLegendre()` method, 324–25
  - Geary, David, 420
  - Generic class libraries, 355–63
    - defined, 355
    - least squares fit (example), 356–58
    - `LineDemo` Class, 361–63
    - problem analysis, 356
    - problem-specific part, implementing, 358–61
    - testing, 361–63
  - `getContentPane()` method, 426
  - `getDensity()` method:
    - `PerfectGas` class, 230
    - `RealGas` class, 235
  - `getEnthalpy()` method, 136
    - `PerfectGas` class, 230
  - `getEntropy()` method, 136
    - `PerfectGas` class, 230
  - `getField()` method, 136

- getFunction() method, 355
    - ODE class, 282, 301
  - getParameter() method, 443
  - getParameter() method, ServletRequest interface, 441
  - getPressure() method, RealGas class, 235
  - getTemperature() method, RealGas class, 235
  - getViscosity() method:
    - PerfectGas class, 230
    - RealGas class, 236, 267–68
  - getWriter() method, 443
  - Glass panes, 420–21
  - Gordon, Rob, 124
  - Graphic Java, Volume 2: Swing (3rd ed.)* (Geary), 420
  - Graphical User Interface (GUI), 3, *See also* GUI libraries
  - Green Project, 2
  - Grid, 306
  - GridBagLayout class, 426
  - GUI libraries, 176–78, 420–21
    - java.awt package, 176
    - java.awt.event package, 176
    - javax.swing package, 177
    - javax.swing.border package, 177
    - javax.swing.event package, 177
    - javax.swing.table package, 177
    - javax.swing.text package, 177
    - javax.swing.tree package, 177
  - GUIs, 419–33
    - AtmGUI class, 421–23
      - final form of, 430–33
    - container:
      - adding components to, 425–27
      - choosing, 422–23
    - event handlers, 427–30
    - GUI components, selecting, 423–25
    - moving from C to Java, 30
    - moving from Fortran to Java, 20
- H**
- HashMap class, 170, 183
  - HashSet class, 170
  - Hashtable, 170
  - Header, 112
  - Hooke's law, 280–81
  - HttpServlet class, 439–40
    - subclass, general form of, 440–41
- I**
- I/O capability, 381–417
    - atmosphere modeling tool (test case), 403–7
    - byte input streams, 384–89
    - byte output streams, 389–96
    - character input streams, 396–99
    - character output streams, 399–403
    - command line arguments, getting input from, 407–9
    - general concepts, 382–83
    - java.nio package, 417
    - moving from C++ to Java, built-in math functions, 35
    - reading to a file, 412–14
    - restoring objects, 414–16
    - saving objects, 414–16
    - streams, using, 409–11
    - writing to a file, 412–14
  - IEEEremainder() method, 209
  - if-else statements, 54–55
  - IfDemo class, 55
  - ImplementDemo class, 135–36
  - import declarations, 142–43
  - Improper integrals, 308
    - solving, 317–22
  - IncopDemo class, 47
  - Increment/decrement operators, 46–47
  - Indices, arrays, 16–17
  - InfiniteChargedPlate class, 134–35
  - Inheritance, 39, 67, 82–84, 111
    - and access modifiers, 84–85
    - and interfaces, 136
    - member hiding/member overriding, 86
    - and method arguments, 85
    - moving from C++ to Java, 34
  - Initializers, classes, 65
  - Input streams, 382–83
  - InputDemo class, 122–23
  - InputStream class, 385
    - close() method, 385
  - InputStreamReader class, 383, 398–99
  - Inside Servlets: Server-Side Programming for the Java Platform (Callaway), 436
  - Instance (inner) nested classes, 77, 78–80
  - Instance methods, 16, 70, 116–18
  - Instance variables, 70, 91, 95–98
  - InstanceDemo class, 117–18
  - int data type, 14–15, 24, 93, 208–9
  - Integer class, 184–86
  - Integral equations, 307–31
    - closed-form solutions, 308–9
    - Fredholm integrals, 308, 326
    - Gaussian quadrature methods, 322–26
    - general form of, 308
    - general integral types, 326
    - improper integrals, 308
      - solving, 317–22
    - open-form solutions, 308
    - proper integrals, 308
    - Simpson's rule, 314–17
    - thin airfoil theory (example), 326–31

- trapezoidal algorithms, 309–14
- Integrator class, 311–12, 319
- Interfaces, 67, 131–38
  - compared to abstract classes, 132
  - declaring, 132–33
  - defined, 131
  - as an example of polymorphism, 40
  - implementing, 134–36
  - and inheritance, 136
  - instances as input parameters and return types, 136–38
  - members, 133
  - moving from C++ to Java, 34
- Internet, 2
- Inverse transforms, 335–36
- InvertDemo class, 264–65
- invertMatrix() method, 261–63
- Ionized interface, 136
- IonizedSpecies class, 84, 141

## J

- JANAF Thermochemical Tables, 370

- jar, 4

- jar utility, 145–49

- options, 145

- jarsigner, 4

- Java:

- access privileges, 67–68

- assignment operators, 14

- basic syntax, 41–64

- comments, 43

- general syntax, 41–43

- keyboard I/O, 62–64

- loops/control structures, 54–60

- operators, 44–54

- printing, 62–64

- simple Java program, 42–43

- braces, 13

- bytecode, 6

- class loader, 7

- classes, 12–13, 65–90

- declaration syntax, 66–67

- code blocks, 13

- comment statements, 13

- compiler, 6

- compiler options, 8

- expandability of, 7

- free-form coding, 13

- future of, 4

- history of, 2–4

- I/O capability, 381–417

- installing, 7–8

- as an interpreted language, 6

- Java APIs, 28–29

- keywords, 15

- loop and flow control structures, 14

- mathematical operators, 14

- memory model, 6

- moving from C to, 21–30

- arrays, 26–27

- basic printing, 30

- basic syntax, 23

- built-in math functions, 29

- C structs, 23

- classes, 22, 23

- dynamic memory allocation, 27–28

- exception handling, 28

- functions, 25–26

- GUIs, 30

- I/O capability, 30

- libraries, 28–29

- main () method, 22

- methods, 25–26

- pointers, 25

- program structure, 22

- strings, 29

- variables, 24–25

- Web-base applications, 30

- moving from C++ to, 31–36

- basic syntax, 32

- built-in math functions, 34–35

- data types, 33

- enumerations, 34

- I/O capability, 35

- inheritance, 34

- interfaces, 34

- memory management, 35–36

- pointers, 33–34

- preprocessor directives, 32–33

- strings, 35

- structures, 34

- unions, 34

- moving from Fortran to, 11–20

- arrays, 16–17

- basic syntax, 13–14

- built-in math functions, 19

- code libraries, 18–19

- dynamic memory allocation, 17

- exception handling, 18

- functions, 15–16

- GUIs, 20

- I/O capability, 19–20

- methods, 15–16

- pointers, 17–18

- program structure, 12–13

- subroutines, 15–16

- variables, 12, 14–15

- Web-base applications, 20

- Java (*cont.*)
    - as multithreaded language, 7
    - object-oriented nature of, 5, 11
    - packages, 19, 28–29, 67
    - platform neutrality of, 5–6
    - primitive types, 15, 16
    - reference types, 15, 16
    - robustness of, 6
    - security of, 6–7
    - simplicity/familiarity of, 5
    - source code, initial release of, 3
    - as a standard, 1
    - as strongly typed language, 15
    - types, 14–15, 24–25
    - versatility of, 7
  - Java 1.0, 2–3
  - Java 1.1, 3
  - Java 1.2, 3–4
  - Java 1.3, 4
  - Java 1.4, 4
  - Java 2 Platform Enterprise Edition (J2EE), 3, 173
    - downloading, 7
    - SDK, 438
  - Java 2 Platform Micro Edition (J2ME), 3, 173
    - downloading, 7
  - Java 2 Platform Standard Edition (J2SE), 3
    - core J2SE libraries, 174–76
      - java.io package, 174–75
      - java.lang package, 174–75
      - java.math package, 174–75
      - java.util package, 175–76
    - GUI libraries, 176–78
      - java.awt package, 176
      - java.awt.event package, 176
      - javax.swing package, 177
        - javax.swing.border package, 177
        - javax.swing.event package, 177
        - javax.swing.table package, 177
        - javax.swing.text package, 177
        - javax.swing.tree package, 177
    - obtaining, 7
  - Java 2 Software Development Kit (SDK), 4
  - Java APIs, 28–29, 69
    - collection classes in, 170–71
  - Java Archive (JAR) files, 139, 145–49
    - creating, 146–47
    - extracting files from, 147–48
    - jar utility, 145–49
    - running an application from, 148
    - viewing the contents of, 147
  - Java Authentication and Authorization Service (JAAS), 4
  - Java class libraries, 173–78
  - Java Cryptography Extension (JCE), 4
  - Java Database Connectivity (JDBC), 3, 174
  - Java Event Handling* (Palmer), 428
  - Java How-to Program* (Deitel/Deitel), 124
  - Java Naming and Directory Interface (JNDI), 4
  - Java Native Interface (JNI), 3, 19
  - Java Print Service API, 4
  - Java programs, compiling/running, 8–9
  - Java Secure Socket Extension (JSSE), 4
  - Java servlets, *See* Servlets:
  - Java Virtual Machine (JVM), 6
    - obtaining, 7
    - options, 9
  - Java Virtual Machine (JVM), 438
  - Java web server, 437–38
  - Java Web Start, 4
  - java.awt package, 176, 420, 426
  - java.awt.event package, 140, 176
  - JavaBeans, 3–4
  - javac, 4, 8
  - javadoc, 43
  - java.io package, 174–75, 381–83, 417
  - java.io.package, 19
  - java.lang package, 174–75, 197–98
  - java.math package, 174–75, 197, 212
  - java.nio package, 417
  - JavaServer Pages (JSPs), 30
  - java.swing.table package, 140
  - java.util package, 137, 159, 175–76
  - javaw, 4
  - javax.swing package, 177
  - javax.swing.border package, 177
  - javax.swing.event package, 177
  - javax.swing.table package, 177
  - javax.swing.text package, 177
  - javax.swing.tree package, 177
  - JButton, 426, 428
  - JButton class, 177
  - JComboBox, 423–24, 428–29
  - JFrame, 427
  - JPanel, 426–27
  - JRadioButton class, 177
  - JSplitPane class, 177
  - JTabbedPane class, 177
  - JTable class, 177–78
  - JTextArea, 424, 426, 429
  - JTextField, 423–24, 429
  - JToggleButton class, 177
  - JTree class, 177–78
- K**
- Keyboard I/O, 62–64
  - Keywords, 15



**L**

Layered panes, 420–21  
Least squares fit to a polynomial equation, 366–68  
leastSquaresFit () method, 354, 361, 363, 366, 368–69, 371, 374  
Length, arrays, 169–70  
length () method, 191  
Levenberg-Marquadt methods, 378  
LexComparator class, 191–92  
Libraries:  
    core J2SE libraries, 174–76  
    generic class libraries, 355–63  
    GUI libraries, 176–78  
    Java class libraries, 173–78  
    Java I/O libraries, 383  
    moving from C to Java, 28–29  
    moving from Fortran to Java, 18–19  
LineDemo Class, 361–63  
LinkedHashMap class, 170  
LinkedHashSet class, 171  
LinkedList class, 171  
loadArrays () method, 368–70, 374  
    Polynomial class, 359–60  
log () method, 203–4, 217  
log10 () method, 216–18  
LogDemo class, 203–4  
logX () method, 216–18  
Long class, 184–86  
long data type, 14–15, 93, 208  
Loops, 54–60  
Lower-upper decomposition (LU), 250, 257–61  
luDecomp () method, 259–61

**M**

Maclaurin series, 216  
main () method, 13, 42, 407–9  
    general syntax, 119  
    using command-line arguments in, 120  
Math class, 197, 198–204  
Math functions, *See* Built-in math functions:  
Math2 class, 126–27  
    compiling, 222  
    final version of, 220–22  
    methods, 222–23  
Mathematical constants, 199  
Matrix inversion, 261–63  
max () method, 208–9  
MaxDemo class, 208–9  
Maximum methods, 208–9  
Mean camber line, 327  
Member hiding/member overriding, and inheritance, 86

**Members:**

    classes, 65  
    interfaces, 133  
Memory management, moving from C++ to Java, 35–36  
META-INF directory, 146  
Method arguments:  
    arrays as, 167–69  
    and inheritance, 85  
Method chaining, 124–25  
Method overloading, 126–27  
Method overriding, 127–29  
MethodArgDemo class, 85  
Methods, 12–13, 111–30  
    abstract, 123–24  
    access modifiers, 114–16  
    chaining, 124–25  
    declaring, 70–71, 112–14  
    final, 124  
    input parameters, 121–23  
    instance, 116–18  
    main () method, 119–20  
    moving from C to Java, 25–26  
    moving from Fortran to Java, 15–16  
    naming conventions/restrictions, 114  
    native keyword, 124  
    overloading, 126–27  
    overriding, 127–29  
    passing arguments to, 121–23  
    return statement, 129–30  
    signature, 40  
    static, 118–19  
    synchronized keyword, 124  
    types of, 16  
MethodsDemo class, 192  
midpoint () method, 319–22  
    source code, 320–21  
    TestMidpoint class, 321–22  
Midpoint rule, 318  
min () method, 208–9  
Minimum methods, 208–9  
Miscellaneous operators, 51–53  
MiscOpDemo class, 52–53  
Monte Carlo algorithms, 378  
Multiline comment, 13

**N**  
N class, 241–42  
N2 class, 240–41  
Naming conventions:  
    methods, 114  
    variables, 101  
native keyword, 124  
Navier-Stokes equations, 271, 298

- Nested classes, 77–81
  - anonymous inner classes, 77, 80–81
  - defined, 77–78
  - instance (inner) nested classes, 77, 78–80
  - static nested classes, 77, 78
- Newton's second law, 281
- NitrogenGas class, 242–45
- Nonpolynomial equations, fitting to, 373–74
- Nonstatic fields, 69–70
- Nyquist sampling frequency, 345–46
- O**
- Oak, 2
- Object-oriented programming, 37–40
  - classes, 37
  - encapsulation, 37–38
  - inheritance, 39, 67
  - interfaces, 67
  - objects, 37–38
  - polymorphism, 40
- ObjectInputStream class, 384, 388–89
- ObjectOutputStream class, 393–95
- Objects, 13, 37–38, 68–69
  - copying, 75–77
- ODE class, 273–75
- ODEshooter () method, 294–96, 303–4
- ODESolver class, 279–80, 356
- One-dimensional arrays, 160–61
- Open formulas, 318
- Operators, 44–54
  - arithmetic operators, 44–45
  - assignment operators, 45–46
  - bitwise operators, 50–51
  - boolean operators, 48–50
  - increment/decrement operators, 46–47
  - miscellaneous operators, 51–53
  - precedence, 53–54
  - relational operators, 47–48
- OpPrecedence class, 53
- Ordinary differential equations (ODEs), 271, 272–73
  - first-order ODE, 272–73
  - second-order ODE, 272
- Output streams, 382–83
- OutputStream class, 390–91
  - close () method, 391
- OutputStreamWriter class, 401–2
- OverrideDemo class, 129
- P**
- Packages, 19, 28–29, 67, 139–48
  - and access control, 144
  - bytecode files of, 140
  - CLASSPATH environment variable, 143–44
  - defining, 140–42
  - definition of, 139
  - import declarations, 142–43
  - including classes in, 141–42
  - user-defined package, importing, 142–43
- Palmer, Grant, 428
- Panes, 420–21
- ParallelPlate class, 135
- parse () methods, 185–86
- ParseDemo class, 195–96
- parseDouble () method, Double class, 195
- Partial differential equations (PDEs), 271, 306
- Partial pivoting, 250–51
- partialPivot () method, 251–54
- Passing arguments to methods, 121–23
- Passing arrays as method arguments, 168–69
- Passing exceptions to methods, 156–57
- PerfectGas class, 227–28, 230–33
  - getDensity () method, 230
  - getEnthalpy () method, 230
  - getEntropy () method, 230
  - getViscosity () method, 230
- PipedInputStream class, 389
- PipedReader class, 399
- PipedWriter class, 403
- Pivoting, 250–53
  - full, 250–51
  - partial, 250–51
  - partialPivot () method, 251–54
- Pointers:
  - moving from C to Java, 25
  - moving from C++ to, Java, 3
- Polymorphism, 40, 111–12
- Polynomial class, 358–61, 369–70
- pow () method, 201–2
- PowDemo class, 202
- Power class, 374–78
- PowerDemo class, 376–77
- Prandtl number, 299
- Precedence, 53–54
  - operators, 53–54
- Preprocessor directives, moving from C++ to, 32–33
- Primitive types, 15, 16
  - converting to strings, 193
  - input arguments, 122–23
- Primitive variable wrapper classes, 179–81
  - constructors, 182
  - objects:
    - converting to a primitive value, 184–85
    - creating, 181–84
    - parse () methods, 185–86

- print () method:
    - PrintStream class, 393
    - PrintWriter class, 402–3
  - printArray () method, 168
  - PrintWriter class, 402–3
  - Printing, 62–64
  - println () method, 42
  - println () method:
    - PrintStream class, 394
    - PrintWriter class, 402–3
  - PrintOutputStream class, 395–96
  - PrintStream class, 390
  - PrintWriter class, 399
  - private access, 67–68
    - variables, 102
  - Program structure:
    - moving from C to Java, 22
    - moving from Fortran to, 12–13
  - Proper integrals, 308
  - protected access, 67–68, 144
    - variables, 102
  - public access, 67–68
    - variables, 102
  - Public interface, 133
  - PushbackReader class, 399
- Q**
- Query string, 441
- R**
- Random number generator methods, 211–12
  - RandomDemo class, 211–12
  - read () method:
    - InputStream class, 385
    - Reader class, 396–97
  - Reader class, 396–97
    - close () method, 396–97
  - readFully () method:
    - DataInputStream class, 387
    - ObjectInputStream class, 388–89
  - readLine () method, 396
    - BufferedReader class, 398–99, 412
  - readObject () method, ObjectInputStream class, 389, 416
  - Real gas viscosity method, 265–69
  - RealGas class, 227–28, 234–38
    - getDensity () method, 235
    - getPressure () method, 235
    - getTemperature () method, 235
    - getViscosity () method, 236
  - RealGasDriver class, 244–45
  - Rectangle class, 176
  - Reference types, 15, 16
    - converting to strings, 193
    - input arguments, 122–23
  - Relational operators, 47–48
  - RelationalOpDemo class, 48
  - Remainder methods, 209–11
  - Remote Method Invocation (RMI), 3
  - RestoreObject class, 416–17
  - return statements, 62, 129–30, 218
  - Return types, arrays as, 167–69
  - ReturnDemo class, 130
  - rint () method, 209
  - rmic, 4
  - round () method, 209
  - RoundDemo class, 210–11
  - Rounding methods, 209–11
  - Runge-Kutta schemes, 276–80, 356
    - orders of accuracy, 277–78
  - rungeKutta4 () method, 279–84, 290
- S**
- Sampling theory, 345–47
    - aliasing, 346–47
  - SaveObject class, 415–16
  - Scope, variables, 108–10
  - ScopeDemo class, 109
  - Second-order ODE, 272
  - SequenceInputStream class, 389
  - Servlets:
    - creating Web-based applications using, 435–48
    - defined, 437
    - and Java web server, 437
    - required libraries/tools, 438
  - setContentPane () method, 426
  - setInitialConditions () class, ODE class, 282, 301
  - setLayout () method, 426
  - Shadowed variables, 108, 121
  - Shape class, 88
  - Shooting, 292
  - ShootingCompress class, 303–5
  - Short class, 184–86
  - short data type, 14–15, 24, 93
  - Signature, methods, 40
  - Simple Java program, 42–43
  - SimpleGas class, 70
  - SimpleGas2 class, 71
  - SimpleGas3 class, 73
  - SimpleGas4 class, 75–76
  - SimpleGas5 class, 94–95
  - SimpleGas6 class, 100–101
  - SimpleProgram class, 42

- Simpson's rule:
    - shared characteristic with trapezoidal algorithm, 315
    - TestSimpson class, 316–17
  - simpsonsRule() method, 315–19, 326, 329
  - source code, 316
  - sin() method, 205
  - Single-line comments, 13
  - size() method, ByteArrayOutputStream class, 391–92
  - Software Development Kit (SDK), 4, 7, 173
    - obtaining, 8
  - SolverDemo class, 263–65
  - Sonine polynomial expansion, 266
  - Species class, 83–84, 238–40
    - class hierarchy, 228
  - Species2 class, 87
  - SpeciesDriver class, 142–43
  - Spectral leakage, 347–49
  - split() method, 191
  - sqrt() method, 201–2
  - SqrtFunction2 class, 321
  - Stack class, 171
  - Static fields, 69–70
  - Static initialization blocks, 74
  - Static methods, 16, 70, 118–19
  - Static nested classes, 77, 78
  - StaticDemo class, 118–19
  - StaticInitDemo class, 74
  - StdIODemo class, 410–11
  - Stream-based I/O system, 20
  - Streams, 381
  - StrictMath class, 197, 198–204
  - String class, 14–15, 20, 186–87
    - concatenating strings, 189–90
    - constructors, 187
    - converting primitive and reference types to strings, 193
    - creating strings, 188–89
    - methods, 190–92
    - string objects, obtaining, 187–89
  - StringBufferInputStream class, 389
  - StringReader class, 399
  - Strings:
    - concatenating, 189–90
    - converting primitive and reference types to, 189–90
    - converting to primitive values, 193–96
    - creating, 188–89
    - moving from C to Java, 29
    - moving from C++ to Java, 35
  - StringWriter class, 403
  - struct type, 24
  - Structures, moving from C++ to Java, 4
  - Subroutines, moving from Fortran to Java, 15–16
  - substring() method, 191
  - super keyword, 86
  - Swing containers, 420–21
  - Swing GUI components, 421
  - Swing GUI libraries, 4
  - Swing packages, 4
  - switch statement, 14
  - switch statements, 59–60
  - SwitchDemo class, 60
  - synchronized keyword, 124
  - synchronized keyword, 133
  - Syntax, 41–64
    - general syntax, 41–43
    - keyboard I/O, 62–64
    - loops/control statements, 54–62
      - do-while loops, 56–57
      - if-else statements, 54–55
      - for loops, 57–58
      - switch statements, 59–60
      - while loops, 55–56
    - moving from C to Java, 23
    - moving from C++ to Java, 32
    - moving from Fortran to, 13–14
    - operators, 44–54
      - arithmetic operators, 44–45
      - assignment operators, 45–46
      - bitwise operators, 50–51
      - boolean operators, 48–50
      - increment/decrement operators, 46–47
      - miscellaneous operators, 51–53
      - precedence, 53–54
      - relational operators, 47–48
    - printing, 62–64
    - simple Java program, 42–43
  - Systems of equations:
    - EqnSolver class, 249
    - methods, testing, 263–65
    - Gauss-Jordan elimination, 253–55
    - Gaussian elimination, 255–57
    - general considerations, 248
    - lower-upper decomposition, 257–61
    - matrix inversion, 261–63
    - pivoting, 250–53
    - real gas viscosity method, 265–69
    - solving, 247–70
    - test case, 249–50
- T**
- tan() method, 205
  - TechJava.Gas package, 229, 231–33

- TechJava.MathLib package, 249, 273, 311, 313, 321
- Ternary operators, 44
- TestComplex class, 344–45
- TestDFT class, 340–41
- TestFFT class, 353–54
- TestGauss class, 325–26
- TestMidpoint class, 321–22
- TestPoint class, 321–22
- TestSimpson class, 316–17
- TestTrap class, 314
- Thermal boundary layer, 298
- Thin airfoil theory (example), 326–31
- ThinAirFoil class, 330
- Thinking in Java* (Eckel), 225
- this keyword, 86–87
- throw and throws keywords, 155–57
- ThrowDemo class, 157
- Tomcat server, 438
- toString() method, 193
- Transcendental math functions, 203–4
- transient keyword, 133
- Transient variables, 106
- Trapezoidal algorithms, 309–14
- Trapezoidal method, 310
- trapezoidal() method, 311–14, 317, 319  
source code, 311–12
- TreeMap class, 171
- TreeSet class, 171
- TrigDemo class, 205
- Trigonometric methods, 204–6
- trim() method, 191
- try keyword, 152
- try statements, 151–55  
catch clause, 152–53  
finally clause, 153  
try keyword, 152  
using, 154–55
- TryDemo class, 154–55
- Two-dimensional arrays, 161–64
- Two-point boundary problems, 292, 305–6
- TwoDimDemo class, 163–64
- Types, 14–15
- U**
- Unary operators, 44
- Under-relaxation procedure, 293–94
- union type, 24
- Unions, moving from C++ to Java, 34
- Universal Gas Constant, 226–27
- USatm76 class, 404–7, 438
- User-defined math functions, 215–24  
basic plan of attack, 216  
comparing Java/C/Fortran values, 223–24  
gamma function, 219–20  
hyperbolic trigonometric methods, 218–19  
logarithm methods, 217–18  
Math2 class, 216–17  
compiling, 222  
final version of, 220–22  
methods, 222–23
- User-defined package, importing, 142–43
- UserMathDemo class, 222–23  
source code, 223
- V**
- valueOf() class, Double class, 185–86
- Variables, 12, 24–25, 91–110  
access modifiers, 102–3  
casting, 106–8  
class, 69–70, 91, 95–98, 99  
creating, 98–101  
defined, 91  
final, 105–6  
free, 273  
initializing, example of, 100–101  
instance, 70, 91, 95–98  
moving from C to Java, 24–25  
moving from Fortran to Java, 14–15  
naming conventions/restrictions, 101  
primitive type, 92–95, 98  
reference type, 92–95, 98  
scope, 108–10  
shadowed, 108, 121  
transient, 106  
values, accessing, 103–5  
visible, 108  
volatile, 106
- Vector class, 171
- ViscosityDemo class, 268–69
- volatile keyword, 133
- Volatile variables, 106
- Volterra integrals, 326
- W**
- Web-base applications:  
moving from C to Java, 30  
moving from Fortran to Java, 20
- Web-based applications:  
AtmServlet class, 443–47  
basics of, 436–37  
client machine, sending output back to, 443  
creating using servlets, 435–48  
deploying, 447–48

- Web-based applications (*cont.*)
  - HttpServlet class, 439–40
    - subclass, general form of, 440–41
  - input parameters, extracting, 441
  - server-based applications, running, 442
  - web-based atmosphere modeling tool, 438
- Weights, 322
- while loops, 55–56
- while statement, 14
- write() method:
  - OutputStream class, 391
  - Writer class, 400
  - writeBytes() method, DataOutputStream class, 393
  - writeChars() method, DataOutputStream class, 393
  - writeObject() method, ObjectOutputStream class, 393, 415
- Writer class, 399–400
  - close() method, 400
  - writeUTF() method, DataOutputStream class, 393