

INDEX

- + (addition operator), 106
- + continued SQL statement directive, 31*t*
- * (multiplication operator), 106
- * (escape character), 95*t*
- / slash (division operator), 106
- 01–99 indicator, 14*t*
- IP indicator, 14*t*, 19

A

- A operation extender, 213*t*
- %ABS (Absolute Value), 120*t*, 129, **129**
- access paths, 542, 549–552
 - files and file processing, 531
 - key lists and, 549–552, **550**, **551**
- accessing files. *See* files and file processing, accessing
- ACQ (Acquire), 205*t*, 228–229, 228*t*, **229**, 332, 364
 - file processing using, 530*t*
 - free-format syntax in, 221*t*
 - syntax diagram for, 215*t*
 - workstation device files and, 552*t*
- ACTGRP parameter and keywords, Create Bound Program (CRTBNDRPG) and, 609
- activation groups, named, 609
- ADA programming language, 443
- ADD (Addition), 205*t*, 222, 230, 230*t*, **230**, 456
 - free-format syntax and, 569–570, **569**, **570**
 - indicator usage in, 13, **13**, , 444
 - indicatorless code using, 14, **14**
 - syntax diagram for, 215*t*
- ADDSDUR (Add Date/Time Duration), 205*t*, 211, 212, 231, 231*t*, **230**
 - data/time manipulation using, free- vs. fixed-format syntax, 580*t*
 - syntax diagram for, 215*t*
- addition, 106
 - ADD, 13, **13**, 14, **14**, 205*t*, 215*t*, 222, 230, 230*t*, **230**, 444, 456, 569–570, **569**, **570**
 - date/time (ADDSDUR), 205*t*, 211, 212, 215*t*, 231, 231*t*, **230**, 580*t*

- zero and add (Z-ADD), 26, 208*t*, 219*t*, 344 , 425, 425*t*, **425**
- %ADDR (Get Address), 120, 120*t*, 129–130, **130**
 - return values from, at compile time, 119
- addresses
 - %ADDR (Get Address), 119, 120, 120*t*, 129–130, **130**
 - procedure, Get (%PADDR), 119, 120, 124*t*, 175–175, **175**
 - subprocedure, Get At Runtime (GetProcAddr), 175
- ALIGN keyword, 61*t*
- *ALL figurative constant, 22*t*, 29*t*, 314*t*, 337, 365
- *ALLG , 23*t*
- ALLOC (Allocate Memory), 205*t*, 232, 232*t*, **232**, 362
 - syntax diagram for, 215*t*
- %ALLOC (Allocate Memory), 120*t*, 262, 362
- *ALLX figurative constant, 22*t*
- ALT keyword, 61*t*
- alternate syntax options, 561–584. *See also* free-format code
- ALTSEQ keyword, 38*t*, 61*t*
- ALWNULL parameter and keywords, 39*t*
 - Create Bound Program (CRTBNDRPG) and, 617–618
- ampersand, 95*t*
- analysis, structured, 462
- AND, 106, 205*t*, 233, 233*t*, **233**, 457
 - %BITAND, 131–132, **131**, **132**, 132*t*
 - conditioning and branching operations using, 451*t*
 - conditioning indicators not supported by, 209
 - DOW and, 275–277, 275*t*, **276**, **277**
 - fixed-format in, 220, 220*t*
 - IF and, 472–473, **473**
 - IF, 302–304, 302*t*, **303**, **304**
 - syntax diagram for, 215*t*
 - WHEN, 402–403, 402*t*, **403**
- AND/OR, 10*t*
- Andrews, Robert, 105
- APIs. *See* system functions/interfaces
- application source file, 435
- application-specific procedures, 510
- applications, hierarchy of, 485
- arguments
 - built-in functions and, 119–120
 - operation codes and, 28, 28*t*

- arrays, 9
- compile-time data and, 10*t*, 100–101, **101**
 - cross foot elements in (XFOOT), 195, **195**, 208*t*, 219*t*, 406, 406*t*, **406**
 - Definition specification for , 100–101, **101**
 - free-format syntax and, 574–577, **575** , 576*t*, **577**, 577*t*
 - index of, naming conventions for, 11*t*
 - indicator, 20
 - limits of, 10*t*
 - lookup (LOOKUP), 16, 122*t*, 207*t*, 217*t*, 154–155, 154*t*, **155**, 311–312, 311*t*, **312**, 376, 377, 444
 - move with (MOVEA), 207*t*, 217*t*, 222, 322–323, **323**, 323*t*, 378, 379
 - naming conventions for, 11*t*
 - number of elements in (%ELEM), 152, **152**
 - scan (SCAN), 16, 157–158, 157*t*, **158**, 208*t*, 218*t*, 218, 371–372, 371*t*, **372**
 - search (element lookup) LOOKUP%, 171
 - sort (SORTA), 186, 208*t*, 218*t*, 220, 220*t*, 221*t*, 381–382, 381*t*, **381**, **382**
 - sub- (SUBARR%), 162, 186, **186**
- AS/400, xxi–xxiv, 21, 461, 486, 523, 532
- ASCEND keyword, 61*t*, 381–382
- assignment, 108–110, **109**, **111**
- EVAL and EVALR in, 108–110, **109**
 - evaluate, right-adjust (EVALR), 105, 108–110, **109**, 116, 119, 206*t*, 220, 220*t*, 221*t*, 288–289, 288*t*, **289**, 570–571, 570*t*, **571**, 571*t*
 - evaluation (EVAL), 4, 105, 108–110, **109**, 119–120, 120*t*, 129–130, **130**, 206*t*, 216*t*, 220, 220*t*, 221*t*, 222, 286–287, 286*t*, **287**, 287*t*, 378, 379, 382, 388, 569–571, **569**, **570**, 570*t*, **571**, 571*t*, 584
 - expressions and, 105
 - l-value in , 108–110, **109**
 - operators for, 106
 - r-value in , 108–110, **109**
- *ASTFILL, 149
- ATOLL(), C runtime library function, 588, 601, **602**
- attention identification bytes, function key, 17–18, 17*t*, **18**, **650**
- AUT parameter and keywords, Create Bound Program (CRTBNDRPG) and, 615–616
- automatic storage (memory), 436–437
- B**
- BASED keyword, 61*t*, 499
- BASIC, 108
- BEGSR (Begin Subroutine), 9, 205*t*, 234, 234*t*, **234**, 341, 373
- conditioning indicators not supported by, 209
 - free-format syntax in, 221*t*
 - syntax diagram for, 215*t*
- binary data type, 12*t*
- binary fields, decimal data, 622
- binary operators, 106
- bindable APIs, 587
- binding, 429, 433–434
- ATOLL and, 601
 - bindable APIs and, 587
 - Create Bound Program (CRTBNDRPG) and, 430, 434, 601, 605–618
 - CRTRPGMOD and, 601
 - program, 429
 - QC2LE binding directory and, 601
 - structured programming and, 466
- bit patterns, test (TESTB), 208*t*, 218*t*, 392–393, 392*t*, **393**
- %BITAND, 120*t*, 131–132, **131**, **132**, 132*t*
- %BITNOT, 120*t*, 133–134, **133**, 134*t*
- BITOFF (Set Bit Off), 205*t*, 235, 235*t*, **235**
- syntax diagram for, 215*t*
- BITON (Set Bit On), 205*t*, 236–237, 236*t*, **236**, **237**
- syntax diagram for, 215*t*
- %BITOR, 120*t*, 134–135, **135**, 135*t*
- bits, bitwise operation
- %BITAND, 131–132, **131**, **132**, 132*t*, 131
 - %BITNOT, 133–134, **133**, 134*t*, 133
 - %BITOR, 134–135, **135**, 135*t*, 134
 - %BITXOR, 136–137, **136**, 137*t*, 136
- nibbles in, 392
- set off (BITOFF), 205*t*, 215*t*, 235, 235*t*, **235**
 - set on (BITON), 215*t*, 236–237, 236*t*, **236**, **237**, 236
 - test (TESTB), 208*t*, 218*t*, 392–393, 392*t*, **393**
- %BITXOR, 120*t*, 136–137, **136**, 137*t*
- blank transparency, 622
- *BLANK, 23*t*
- *BLANKS, 23*t*
- blanks, 95*t*
- BLOCK keyword, 48*t*
- BNDDIR parameter and keywords, 39*t*
- Create Bound Program (CRTBNDRPG) and, 609
- Boolean expressions, 108
- Boolean operation codes, 220, 220*t*, 470, 470*t*
- IF, 302–304, 302*t*, **303**, **304**
 - OR, 340, 340*t*, **340**
 - WHEN, 402–403, 402*t*, **403**
- bound procedures and programs, 433–434

- call (CALLB), 205*t*, 215*t*, 241–242, 241*t*, **242**, 343, 344, 345, 347, 429, 433–435, 520, 587, 590–592, 591*t*, **591**. *See also* prototyped calls/procedures
- call, with prototype (CALLP), , 105, 113, 119, 205*t*, 220, 220*t*, 221*t*, 243, 243*t*, **244**, 429, 435, 584, 587, 588–592, 589*t*, **589**, **590**, 591*t*, **591**. *See also* prototyped calls/procedures
- Create Bound Program (CRTBNDRPG) and, 430, 434, 605–618
- brackets, in programming code, 214
- branching, 451–452, 451*t*, **452**, 478–479, 479*t*
- compare and branch (CAB), 205*t*, 215*t*, 220, 220*t*, 238, 238*t*, **238**, 284–285, 284*t*, **285**, 389, 389*t*, **389**, 451*t*, 467*t*, 478–479, 479*t*
- EXSR and, 206*t*, 216*t*, 221*t*, 234, 249–295, 294*t*, **295**, **296**
- GOTO, 301, 301*t*, **301**
- indicators used in, 445–446, **445**, **446**
- operations used in, 479*t*
- built-in functions, 117–200, 120–128*t*
- arguments in, 119–120
- compile-time return values and, 119
- Definition specification and, 120
- duration type, 148, **148**
- File specification and, 120
- reserved words in, 27, 27*t*
- syntax of, 119–120
- BY infix operator, 469
- C**
- C/C++ language, xxii, 3, 6, 37, 108, 443
- ATOLL(), C runtime library function in, 601, **602**
- integer data type capacity limit in, 57
- strings, get/store null-terminated C-language based (%STR), 185, **185**
- CAB (Call and Branch), 205*t*, 238, 238*t*, **238**, 467*t*, 478–479
- branching using, 479, 479*t*
- conditioning and branching operations using, 451*t*
- ENDSR and, 9, 11, 206*t*, 209, 216*t*, 221*t*, 284–285, 284*t*, **285**, 234, 301, 310, 341, 373, 478, 479, 479*t*
- fixed-format in, 220, 220*t*
- syntax diagram for, 215*t*
- TAG and, 9, 11, 208*t*, 209, 218*t*, 389, 389*t*, **389**, 447, 478, 479, 479*t*
- Calculation specification, 3, 4, 7–8, 7*t*, 37, 37*t*, 78, 78–81*t*, 203
- data types supported by, 12
- /END-FREE directive in, 112
- enhanced or alternate syntax for, 563, 564–564, **564**
- expressions and, 112–113, **113**
- free-format, 565–570
- /FREE-FORM directive and, 112
- standard vs. alternate, 78
- calculation statements, 4, 9
- CALL (Call Program), 205*t*, 239–240, 239*t*, **239**, **240**, 343, 344, 345, 347, 433–434, 435
- indicator use and, 444
- QtmhRdSTin (Read from Standard Input/Stdin) and, 594–595, 594*t*, **594**, **595**
- syntax diagram for, 215*t*
- CALLB (Call Bound Program), 205*t*, 241–242, 241*t*, **242**, 343, 344, 345, 347, 429, 433–435, 520. *See also* prototyped calls/procedures
- CGI APIs and, 587
- QtmhPutEnv (put environment value) and, 590–592, 591*t*, **591**
- syntax diagram for, 215*t*
- calling, 433–434, 484, 486–487, **486**, 510–512. *See also* parameter passing
- bound procedures/programs (CALLB), 205*t*, 215*t*, 241–242, 241*t*, **242**, 343, 344, 345, 347, 429, 433–435, 520, 587, 590–592, 591*t*, **591**. *See also* prototyped calls/procedures
- CEEDOD (Retrieve Operational Descriptor) and , 498, 524, **524** , **525**
- data structures passed as parameters in, 493–494, **493**, **494**
- declaring parameters for (PARAM), 207*t*, 217*t*, 209, 343–346, 343*t*, **345**, **346**, 491–493, 588, 590–592, 591*t*, **591** , 594–595, 594*t*, **594**, **595**
- dynamic program call (CALL), 205*t*, 215*t*, 239–240, 239*t*, **239**, **240**, 343, 344, 345, 347, 433–434, 435, 444, 594–595, 594*t*, **595**, **595**
- expression-based, for procedures, 511, **511**
- indicator use and, 444
- modular programming and, 484
- OPDESC keyword and, 523–524
- parameter movement in, 491–493, **492**, **493**, 493*t*
- parameter passing in, 487–493, **493**
- PLIST and, 9, 207*t*, 209, 345, 346–348, 346*t*, **347**, 348*t*, 487–493, **493**
- procedures, subprocedures, subroutines and, 510–512
- prototyped calls in (CALLP), , 105, 113, 119, 205*t*, 220, 220*t*, 221*t*, 243, 243*t*, **244**, 429, 435, 584, 587, 588–592, 589*t*, **589**, **590**, 591*t*, **591**. *See also* prototyped calls/procedures
- qualified program names and, 240
- return to calling program (RETURN), 16, 105, 113, 119, 218*t*, 220, 220*t*, 221*t*, 208*t*, 368–369, 368*t*, **369**
- system functions in, 498

- CALLP (Call with Prototype), 105, 113, 205*t*, 243, 243*t*, **244**, 429, 435. *See also* prototyped calls/procedures built-in functions and, 119
- CGI APIs and, 587
- free-format syntax in, 221*t*, 584
- natural expression support in, 220, 220*t*
- QtmhGetEnv (Get Value Of Environment Variable) and, 588–590, 589*t*, **589**, **590**
- QtmhPutEnv (Put Environment Value) and, 590–592, 591*t*, **591**
- syntax diagram for, 215*t*
- *CANCL, 29*t*
- CAS (CASxx) (Compare and Assign Subroutine), 21, 205*t*, 234, 245–247, 245*t*, **245**, **246**, 467*t*
- conditioning and branching operations using, 451*t*
- END, 283, 283*t*, **283**
- fixed-format in, 220, 220*t*
- indicator use and, 444
- syntax diagram for, 215*t*
- case sensitivity, in programs, 214
- CASE structures, 5, 467, 474–479, **474**
- compare and branch (CAB) in, 205*t*, 215*t*, 220, 220*t*, 238, 238*t*, **238**, 284–285, 284*t*, **285**, 389, 389*t*, **389**, 451*t*, 467*t*, 478–479, 479*t*
- in-line form of, 5, 474–476, **474**
- logic flow in, 474–475, **475**
- logic module controlled by, 476–478, **477**
- nesting and, 474
- otherwise (OTHER), 341, 341*t*, **341**
- SELECT-WHEN and, 475, **475**
- subroutines form of, 474–476, **474**
- successive use of, 476–479, **477**
- top-down design and, 476
- CAT (Concatenate), 205*t*, 247–248, 247*t*, **248**
- syntax diagram for, 215*t*
- CCSID keyword, 39*t*, 61*t*
- *CDMY, 141*t*, 210*t*, 317, 318*t*, 327, 327*t*
- CEEDOD (Retrieve Operational Descriptor), 498, 524, **524**, **525**, **649**
- alternative naming for, 524, **524**, **525**
- CEEGSI (Get String Information), 648, **648**
- CGI application programming interface (APIs), xxiii, 143, 587–600, 587*t*. *See also* Web programming interfaces
- CGI RPG, xxiii
- CHAIN (Random Read/Access), 19, 205*t*, 224, 249–251, 249*t*, **250–251**, 268, 533, **533**, 553, **553**
- cursor position in file after execution of, 543*t*, **545**
- decimal data errors and, 623
- file processing using, 530*t*
- %FOUND versus indicators in, 157–158, 157*t*, **158**
- free-format syntax in, 221*t*, 563, **563**
- key list in (%KDS), 123*t*, 128, 168–169, **169**
- keylists and KLIST in, 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308**, 530*t*
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- syntax diagram for, 215*t*
- workstation device files and, 552*t*
- %CHAR, 121*t*, 137–138, **138**, 571–572, **571**, **572**, 582
- character data, 12*t*, 22–27. *See also* string data
- CEEGSI (Get String Information), 648, **648**
- character literal values, 22–27, 23*t*
- %CHAR, 121*t*, 137–138, **138**, 571–572, **571**, **572**, 582
- clear (CLEAR), 205*t*, 215*t*, 221*t*, 256–258, 256*t*, 257*t*, **258**, 317, 366, 425, 626
- concatenate (CAT), 205*t*, 215*t*, 247–248, 247*t*, **248**
- convert to double-byte value (%GRAPHIC), 158, **158**
- convert to packed decimal (%DEC), , 121*t*, 142–144, **143**, **144**, **573–574**, 574, 588
- convert to packed decimal (%DECH), 121*t*, 142–144, **143**, **144**
- convert to time value (%TIME), 126*t*, 189, 189*t*, 572–573, **573**, 581, 582–583, **582**, **583**
- convert to timestamp (%TIMESTAMP), 126*t*, 190, 190*t*, 581, 582–583, **582**, **583**
- convert to unsigned integer (%UNS), 194–195, **195**
- convert to unsigned integer (%UNSH), 194–195, **195**
- Convert to Uppercase source code for, **646**
- convert to UCS2 (%UCS2), 193
- display message (DSPLY), 278–279, 278*t*, **279**
- edit (%EDITW), 151, **151**
- edit to (%EDITC), 149–150, **150**
- edit to floating point (%EDITFLT), 151
- get/store null-terminated C-language based (%STR), 185,
- move with array (MOVEA), 207*t*, 217*t*, 222, 322–323, **323**, 323*t*, 378, 379
- replace/insert string (%REPLACE), 178
- Return as Uppercase source code for, **645**
- Return Length of Input String source code for, **647**

- scan string of (%SCAN), , 125*t*, 179, **179**
- set length of (%LEN), 119, 124*t*, 170, **170**
- substring then move left (SUBST), 218*t*, 387–388, 387*t*, **388**
- test numeric field for (TESTN), 208*t*, 218*t*, 394, 394*t*, **394**
- test result field of (TEST), 16, 208*t*, 209, 218*t*, 221*t*, 390–391, 391*t*, **391**, 580*t*
- translate (%XLATE), 127*t*, 196, **196**
- translate string (XLATE), 208*t*, 219*t*, 407–409, 407*t*, **408**, **409**
- trim (%TRIM, %TRIML, %TRIMR), 120, 126*t*, 192–193, **192–193**
- verification lists and, 138–139
- verify string (CHECK), 157–158, 157*t*, **158**, 205*t*, 215*t*, 251–253, 251*t*, **252**, **253**
- verify string (%CHECK), 121*t*, 138–139, **139**, 140, **140**
- verify string right to left (CHECKR), 157–158, 157*t*, **158**, 205*t*, 215*t*, 254–255, 254*t*, **255**
- verify string right to left (%CHECKR), 140, **140**
- ZTools source code listings for, 643–650
- character literal values, 22–27, 23*t*
- CHECK (Verify String), 205*t*, 251–253, 251*t*, **252**, **253**
- %FOUND versus indicators in, 157–158, 157*t*, **158**
- syntax diagram for, 215*t*
- %CHECK (Verify String), 121*t*, 138–139, **139**, 140, **140**
- CHECKR (Verify String Right to Left), 205*t*, 254–255, 254*t*, **255**
- %FOUND versus indicators in, 157–158, 157*t*, **158**
- syntax diagram for, 215*t*
- choice, in structured programming, 466, 467, 467*t*
- CL language, 108
- CLASS keyword, 61*t*
- CLEAR (Clear Values), 205*t*, 256–258, 256*t*, 257*t*, **258**, 317, 366, 425
- decimal data errors and, 626
- free-format syntax in, 221*t*
- initial values and 257–258, 257*t*
- syntax diagram for, 215*t*
- CLOSE (Close File), 205*t*, 250, 259, 259*t*, **259**, 297, 553–554, **554**
- free-format syntax in, 221*t*
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- syntax diagram for, 215*t*
- workstation device files and, 552*t*
- *CMDY , 210, 210*t*, 317, 318*t*
- COBOL, xix–xxii, 3, 6, 37, 108
- CodeStudio, 6
- Columbia Dictionary of Quotations, The*, 105
- columnar expression format (hybrid free-format) code, 3
- columns (database). *See* fields
- comments, free-format syntax and, 565–566, **565**
- COMMIT (Commit Transaction), 205*t*, 260–261, 260*t*, **261**, 370, 530*t*
- file processing using, 530*t*
- free-format syntax in, 221*t*
- syntax diagram for, 215*t*
- COMMIT keyword, 48*t*
- COMP (Compare), 16, 119, 205*t*, 215*t*, 261, 261*t*, 444, **261**, 452–454, **452**, **453**, 457, 467*t*
- avoiding use of, 467
- built-in functions and, 119
- conditioning and branching operations using, 451*t*
- indicator use and, 444
- syntax diagram for, 215*t*
- testing for ranges, lists, select/omit using, 453–454, **453**, **454**
- comparison, 451–452, 451*t*, **452**
- compare (COMP) and, 16, 119, 205*t*, 215*t*, 261, 261*t*, 444, **261**, 444, 451*t*, 452–454, **452**, **453**, 457, 467, 467*t*
- compare and branch (CAB), 205*t*, 215*t*, 220, 220*t*, 238, 238*t*, **238**, 284–285, 284*t*, **285**, 389, 389*t*, **389**, 451*t*, 467*t*, 478–479, 479*t*
- compare and perform subroutine (CAS), 21, 205*t*, 215*t*, 220, 220*t*, 234, 245–247, 245*t*, **245**, **246**, 283, 283*t*, **283**, 444, 451*t*, 467*t*
- expressions and, 105, 110–111, **110**
- IF, 302–304, 302*t*, **303**, **304**
- IF-THEN-ELSE in, 470–474
- lists in, 453–454, **453**, **454**
- operators for, 106
- ranges in, 453–454, **453**, **454**
- relationship testing in, 472–474, **472**, **473**
- select/omit in, testing for, 453–454, **453**, **454**
- compile-time data, 100–101, **101**
- arrays and, 10*t*, 100–101, **101**
- limits to, 10*t*
- return values and built-in functions, 119
- tables and, 100–101, **101**

- compiler, xxi, 603–618
 - conditional compiler preprocessor source directives for, 32–33, 33*t*
 - Create Bound Program (CRTBNDRPG) and, 430, 605–618
 - Create RPG Module (CRTRPGMOD) and, 430, 605
 - directives for, , 30–33, 30–31*t*, **31–32**. *See also* directives free-format syntax and, 566–567, **567**, **569**
 - source files and, 433
 - specification sequence and, 8, 37, 37*t*
 - components of RPG, 9
 - concatenation (CAT), 205*t*, 215*t*, 247–248, 247*t*, **248**
 - conditional operations
 - ELSE as, 281, 281*t*, **281**
 - ELSEIF as, 282, 282*t*, **282**
 - IF as, 302–304, 302*t*, **303**, **304**
 - OR as, 340, 340*t*, **340**
 - WHEN as, 402–403, 402*t*, **403**
 - conditioning indicators, 19, 21, 203, 209, 209*t*
 - CONST keyword, 24, 61*t*
 - expressions and, 113
 - %HANDLER and, 123*t*, 128, 159–161, **160**, **161**, 162–166, **163**, 165*t*, 409, 410, 415–418, **416**
 - parameters and, 114–115, **114**, **115**
 - prototyped parameters and, 515*t*
 - constants, 21–27
 - CONST keyword and, 24
 - figurative, 21–27
 - literal values and, 21–27
 - named, 21–27, **25**
 - constructs, structured programming and, 466–469
 - CONTENT_LENGTH , 592, 593*t*, 595
 - continuation, expression, 108
 - control codes, 94, 95*t*
 - control flow constructs, structured programming and, 466–469
 - Control specification, 7–8, 7*t*, 37, 37*t*
 - control values, reserved words in, 27, 27*t*
 - control-level indicators, 18–19, 203
 - controlling indicators, 19
 - Convert to Uppercase source code, **646**
 - COPY, indicator use and, 444
 - COPYNEST keyword, 39*t*
 - COPYRIGHT keyword, 39*t*
 - /COPY directive, 30*t*
 - Create Bound Program (CRTBNDRPG), 430, 434, 605–618
 - ACTGRP parameter and keywords for, 609
 - ALWNULL parameter and keywords for, 617–618
 - AUT parameter and keywords for, 615–616
 - BNDDIR parameter and keywords for, 609
 - CVTOPT parameter and keywords for, 613
 - DBGVIEW parameter and keywords for, 611–612
 - DFACTGRP parameter and keywords for, 607–608
 - FIXNBR parameter and keywords for, 616–617
 - GENLVL parameter and keywords for, 607
 - INDENT parameter and keywords for, 613
 - keywords used with, 605
 - LANGID parameter and keywords for, 614
 - OPTIMIZE parameter and keywords for, 612
 - OPTION parameter and keywords for, 609–611
 - OUTPUT parameter and keywords for, 612
 - parameters for, 605
 - PGM parameter and keywords for, 606
 - REPLACE parameter and keywords for, 614–615
 - SRCFILE parameter and keywords for, 606
 - SRCMBR parameter and keywords for, 606
 - SRTSEQ parameter and keywords for, 614
 - TEXT parameter and keywords for, 607
 - TGTRLS parameter and keywords for, 617
 - TRUNCNBR parameter and keywords for, 616
 - USRPRF parameter and keywords for, 615
 - Web (CGI) programming interfaces and, 601
 - Create Program (CRTPGM), 434
 - Create RPG Module (CRTRPGMOD), 430, 434, 510, 605
 - Web (CGI) programming interfaces and, 601
 - cross-reference, in files and file processing, 531
 - CRTBNDRPG. *See* Create Bound Program
 - CRTRPGMOD. *See* Create RPG Module
 - CTDATA keyword, 62*t*, 100
 - currency
 - control codes in, 94, 95*t*
 - symbol for, 95*t*, 99, **99**
 - cursor positioning in file, after successful operation, 543*t*
 - CURSYM keyword, 40*t*
 - *CURSYM, 149
 - CVTOPT parameter and keywords, Create Bound Program (CRTBNDRPG) and, 613
 - cycle, RPG. *See* RPG cycle
 - *CYMD , 141*t*, 210, 210*t*, 317, 318*t*, 327, 327*t*
- ## D
- D operation extender, 213*t*
 - data
 - clear (CLEAR), 205*t*, 215*t*, 221*t*, 256–258, 256*t*, 257*t*, **258**, 317, 366, 425, 626
 - copy (MOVE), 3, 4, 20, 207*t*, 209, 210, 212, 217*t*, 222, 316–319, 316*t*, 318*t*, **318**, **319**, 378, 379, 444, 568, **569**, 570–571, 570*t*, **571**, 571*t*, 580*t*

- copy, left-justified (MOVE), 207*t*, 209, 212, 271*t*, 281, 324–328, 324*t*, **325**, **326**, 327*t*, **328**, 344, 568, **569**, 570–571, 570*t*, **571**, 571*t*
- force end of (FEOD), 206*t*, 216*t*, 221*t*, 250, 297, 297*t*, **297**, 351–361, 530*t*, 543*t*, **546**
- data areas, 501–505, **501**
- data structure in, 501–502, **502**
- declaration for, 501, **501**
- define (DEFINE), 33*t*, 205*t*, 209, 215*t*, 263–265, 263*t*, **264**, **265**, 503–505, **504**, **505**
- DTAARA keyword and, 501
- *DTAARA, 28*t*, 305, 342, 398, 503
- EXTNAME keyword, 502
- host variables in, 503–505, **504**
- output to external (OUT), 207*t*, 217*t*, 221*t*, 263, 342, 342*t*, **342**, 501, 503–505, **504**, **505**
- post device information to (POST), 207*t*, 217*t*, 221*t*, 349–350, 349*t*, **350**, 530*t*, 552*t*, 555, **555**
- PREFIX keyword, 502
- read in a (IN), , 206*t*, 216*t*, 221*t*, 263, 305, 305*t*, **305**, 501, 503–505, **504**, **505**
- special device files and, 348
- unlock (UNLOCK), 208*t*, 218*t*, 221*t*, 250, 263, 292, 398–399, 398*t*, **399**, 501, 503–505, **504**, **505**, 530*t*, 543*t*
- data manipulation language (DML), 20–21
- data structures, 9
- clear (CLEAR), 256–258, 256*t*, 257*t*, **258**
- convert to (QtmhCvtDB) and, 596–600, 598–599*t*, **599**, **600**
- data areas and, 501–502, **502**
- DIM keyword and, 576, 576*t*
- free-format syntax and, 574–577, **575**, 576*t*, **577**, 577*t*
- input/output feedback data structure (INFDS) in, 59, 59–60*t*
- key list from (%KDS), 123*t*, 128, 168–169, **169**
- limits to, 10*t*
- multiple-occurrence, 574–577, **575**, 576*t*, **577**, 577*t*
- naming conventions for, 11*t*
- number of elements in (%ELEM), 152, **152**
- occurrence index of, 334
- parameter passing and, 498
- passing as parameters, 493–494, **493**, **494**
- program status data structure (PSDS), 53, 58, 58*t*, 555, **650**
- prototyped calls/procedures and, 498–500, **500**
- QUALIFIED keyword and, 576, 576*t*
- referencing , 576, 577*t*
- set/get occurrence of (%OCCUR), 173, **173**
- set/get occurrence of (OCCUR), 334–336, 334*t*, **335**, **336**, 336*t*
- data types, 12, 12*t*
- converting, using free-format syntax, 571–577
- database records and files, 531, 532. *See also* files and file processing; records
- commit change (COMMIT), 205*t*, 215*t*, 221*t*, 260–261, 260*t*, **261**, 370, 530*t*
- dates and times in, 633–634
- define key list for (KLIST), 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308**, 530*t*
- delete record (DELETE), 88, 157–158, 157*t*, **158**, 168–169, **169**, 205*t*, 215*t*, 221*t*, 250, 266–268, 266*t*, **267**, **268**, 308, 308*t*, 355–361, 530*t*, 543*t*, **545**
- fields and, 623
- flat-file systems and, 429–431, 623
- key field in (KFLD), 206*t*, 209, 217*t*, 307, 307*t*, **307**, 530*t*
- read next record with equal key (READE), 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- read prior record from (READP), 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
- read prior record with equal key (READPE), 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- record random access/read of (CHAIN), 19, 157–158, 157*t*, **158**, 168–169, **169**, 205*t*, 215*t*, 221*t*, 224, 249–251, 249*t*, **250–251**, 268, 308, 308*t*, **308**, 351–361, 530*t*, 533, **533**, 553, **553**, 543*t*, **545**, 553*t*, 563, **563**, 623
- relational, 623
- rollback changes (ROLBK), 208*t*, 218*t*, 221*t*, 370, 370*t*, **370**, 530*t*
- unlock data areas/records (UNLOCK), 208*t*, 218*t*, 221*t*, 250, 263, 292, 398–399, 398*t*, **399**, 501, 503–505, **504**, **505**, 530*t*, 543*t*
- update record (UPDATE), 88, 156, **156**, 208*t*, 219*t*, 221*t*, 250, 400–402, 400*t*, **401**, **402**, 530*t*, 543*t*, **548**, 552*t*557–558, **558**
- write records (WRITE), 88, 208*t*, 219*t*, 221*t*, 404–405, 404*t*, **405**, 529, 530*t*, 543*t*, **548**, 552*t*, 554–555, **554**, 558, **559**
- %DATE, 121*t*, 141–142, 141*t*, **141**, 572–573, **573**, 581, 582–583, **582**, **583**
- *DATE, 26*t*, 27*t*, 100*t*
- DateDiff RPG routine, 641*t*
- DATEDIT keyword, 40*t*
- dates, 12*t*
- add duration to (ADDDUR), 231, 231*t*, **230**
- %CHAR, 121*t*, 137–138, **138**, 571–572, **571**, **572**, 582
- convert to date value (%DATE), 121*t*, 141–142, 141*t*, **141**, 572–573, **573**, 581, 582–583, **582**, **583**

dates, *continued*

- convert to packed decimal (%DEC), , 121*t*, 142–144, **143**, **144**, **573–574**, 574, 588
- convert to packed decimal (%DECH), 121*t*, 142–144, **143**, **144**
- DateDiff RPG routine for, 641*t*
- DASEP (separator selection) keyword and, 635
- DATFMT keyword for, 634–635, 634*t*
- DDS layout for, 631–641
- default values for, 633
- difference between (%DIFF), 122*t*, 146, 146*t*, **146**, 211, 580–582
- display file support for, 638–639
- DTStoTM RPG routine for, 639*t*
- duration built-in functions for, 148, **148**
- duration codes for, , 146, 146*t*, 187, 211–212, 211*t*, 296
- edit codes for, 99, 99–100*t*
- external vs. internal format in, 635
- extract portion of (EXTRCT), 206*t*, 211, 212, 216*t*, 580*t*
- extract portion of (%SUBDT), 126*t*, 187, 211, 296, 580
- fixed separators in, 635
- format codes for, 141–142, 141*t*, 209–210, 210*t*, 211*t*, , 318, 327*t*, 631–641, 634*t*
- free-format syntax and, processing using, 579–582, **580**, **581**
- GetDate RPG routine for, 640*t*
- GetDateAsString RPG routine for, 641*t*
- GetDayCount RPG routine for, 640*t*
- GetDayEx RPG routine for, 639*t*
- GetDayName RPG routine for, 639*t*
- GetDayOfWeek RPG routine for, 639*t*
- GetDayOfYear RPG routine for, 640*t*
- GetEndOfMonth RPG routine for, 639*t*
- GetWeek RPG routine for, 640*t*
- GetWeekISO RPG routine for, 640*t*
- length of field for, 633–634
- MAPVAL keyword for, 638–639
- MOVEL and , 207*t*, 209, 212, 271*t*, 281, 324–328, 324*t*, **325**, **326**, 327*t*, **328**, 344, 568, **569**, 570–571, 570*t*, **571**, 571*t*
- retrieve from system (TIME), 396–397, 396*t*, 397*t*, **397**
- retrieving value of, at runtime , 582–583, **582**, **583**
- RPG xTools routines for, 639–641, 639–641*t*
- separator (edit character) for, 634–635
- special values (*HIVAL and *LOVAL) in 637–638, 638*t*
- storage of, 633–634
- substring of (%SUBDT), 126*t*, 187, 211, 296, 580
- subtract duration (SUBDUR), 208*t*, 211, 212, 218*t*, 385–386, 385*t*, **386**, 580*t*
- Super Julian, 633
- test result field of (TEST), 16, 208*t*, 209, 218*t*, 221*t*, 390–391, 391*t*, **391**, 580*t*
- TimeToTM RPG routine for, 639*t*
- TMToDate RPG routine for, 640*t*
- TMtoDTS RPG routine for, 640*t*
- DASEP (separator selection) keyword, 635, 638
- DATFMT keyword, 40*t*, 48*t*, 62*t*, 634–635, 634*t*, 638
 - prototyped parameters and, 515*t*
- *DAY , 26*t*, 27*t*, 27
- *DAYS, 146*t*, 187, 211*t*, 231, 296
- %DAYS, 121*t*, 148
- DBGVIEW parameter and keywords, Create Bound Program (CRTBNDRPG) and, 611–612
- DDS
 - date and time format considerations in, 631–641
 - define file using, 539, **539**, 549, **549**, 551, **551**
 - define physical file using , 532, **532**
 - join logical files and, 540–541, **540**, **541**
- DEALLOC (Deallocate Memory), 205*t*, 262, 262*t*, **262**
 - free-format syntax in, 221*t*
 - syntax diagram for, 215*t*
- %DEC, 121*t*, 142–144, **143**, **144**, **573–574**, 574, 588
- DECEDIT keyword, 41*t*
- %DECH, 121*t*, 142–144, **143**, **144**
- decimal data errors, 620–629. *See also* decimal notation
 - binary and zoned fields in, 622
 - blank transparency and, 622
 - CLEAR operation and, 205*t*, 215*t*, 221*t*, 256–258, 256*t*, 257*t*, **258**, 317, 366, 425, 626
 - conditions for, 628–629
 - database file models and, 623
 - FIXNBR parameter for, 621–623, 629
 - input operation causing, 628–629
 - packed, 620, 626–627, 627*t*, 628*t*
 - sign or status in, 624
 - zoned, 620, 624–626, **624**, 625*t*
- decimal notation, 95*t*. *See also* decimal data errors
 - control codes in, 94, 95*t*
 - decimal data errors and, 619–629
 - edit codes and, 97–98, 97–98*t*
 - programs, position of, 203
 - retrieve positions (%DECPOS), 119, 121*t*, 145, **145**
- declarations, expressions and, 105
- %DECPOS, 119, 121*t*, 145, **145**
 - return values from, at compile time, 119
- DECPREC keyword, 41*t*, 115–116, 115
- DEFINE (Define Parameters), 33*t*, 205*t*, 263–265, 263*t*, **264**, **265**
 - conditioning indicators not supported by, 209

- data areas and, 503–505, **504**, **505**
 syntax diagram for, 215*t*
- Definition specification, 4, 7–8, 7*t*, 37, 37*t*, 53, **53**, 53–69*t*
- arrays using, 100–101, **101**
- built-in functions and, 120
- data types supported by, 12
- expressions and, 112, **112**
- extended name, **53**, 55–56, 56*t*, **56**
- field definition specification in, **54**
- input/output feedback data structure (INFDS) in, 59, 59–60*t*
- integer variables and, 56–58, 57*t*, 58*t*
- keywords for, 61–69*t*
- procedures, subprocedures, subroutines and, definition of, 512–514
- program status data structure (PSDS) and, 53, 58, 58*t*
- special reserved position (star position) specification in, **54**
- tables using, 100–101, **101**
- DELETE, 88, 205*t*, 250, 266–268, 266*t*, **267**, **268**
 cursor position in file after execution of, 543*t*, **545**
 file processing using, 530*t*
 %FOUND versus indicators in, 157–158, 157*t*, **158**
 free-format syntax in, 221*t*
 key list in (%KDS), 123*t*, 128, 168–169, **169**
 keylists and KLIST with, 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308**, 530*t*
 READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
 READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
 syntax diagram for, 215*t*
- DESCEND keyword, 62*t*, 381–382
- design
 program, 462
 structured, 462
- desk coding, 6
- *DETC, 29*t*
- *DETL, 29*t*
- device files
 force next input from (NEXT), 207*t*, 217*t*, 221*t*, 332–333, 332*t*, **333**
 indicators and, 16
 post device information (POST), 207*t*, 217*t*, 221*t*, 349–350, 349*t*, **350**, 530*t*, 552*t*, 555, **555**
 release (REL), 207*t*, 218*t*, 221*t*, 292, 530*t*, 552*t*
 SPECIAL, 347–348, 348*t*
 workstation, processing of, 551–555, 552*t*
- DEVID keyword, 48*t*, 332
- DFTACTGRP keyword, 41*t*
- Create Bound Program (CRTBNDRPG) and, 607–608
- DFTNAME keyword, 41*t*
- %DIFF, 122*t*, 146, 146*t*, **146**, 211, 580–582
- DIM keyword, 25, 62*t*, 100, 334
 free-format syntax and, 576, 576*t*
 %HANDLER and, 123*t*, 128, 159–161, **160**, **161**, 162–166, **163**, 165*t*, 409, 410, 415–418, **416**
- directives, 30–33, 30–31*t*, **31–32**
 conditional compiler preprocessor source, 32–33, 33*t*
 free-format syntax and, 566, **566**
- directories and subdirectories, 430–431
 qualified program names and, 240
- DISK files, 84, 532
- display files, date support in, 638–639
- display station. *See* workstations
- DIVIDE (Divide), 16, 205*t*, 222, 269–270, 269*t*, **270**, 456
 free-format syntax and, 569–570, **569**, **570**
 indicator use and, 444
 syntax diagram for, 215*t*
- %DIV, 122*t*, 147, **147**
- division, 106
 %DIV, 147, **147**
 DIVIDE, 16, 205*t*, 222, 269–270, 269*t*, **270**, 456
 move remainder of division (MVR), 207*t*, 217*t*, 331, 331*t*, **331**
 remainder (%REM), 177, **177**
- *DMY *t*, 141*t*, 210, 210*t*, 318*t*, 327, 327*t*, 634*t*
 separator for, 635
- DO, xxi, 16, 21, 205*t*, 271–273, 271*t*, **271**, **272**, **273**, 274, 448–450, **448**, **449**, **450**, 465, 467–469, **468**, 468*t*, **469**.
See also DOU; DOW
- branching operations and, 479, 479*t*
- conditioning and branching operations using, 451*t*
- DOU and, 450, **450**. *See also* DOU
- DOW and, 450, **450**. *See also* DOW
- end marker for, 272
- END, 283, 283*t*, **283**
 indicator use and, 444, 445
- %INT and, 123*t*, 167–168, **168**, 588
- ITER, 206*t*, 306, 216*t*, 221*t*, 306*t*, **306**, 451*t*, 468*t*, 479, 479*t*
- leave (LEAVE), 207*t*, 217*t*, 221*t*, 299, 309, 309*t*, **309**, 451*t*, 468*t*, 479, 479*t*
- level-break processing and, 449–450, **449**
- nesting in, 10*t*, 227
- syntax diagram for, 216*t*
- DOU, 21, 105, 128, 205*t*, 206*t*, 273–274, 273*t*, **274**, **275**, 443, 450, **450**, 455, 467–469, **468**, 468*t*, **469**
 AND, 233, 233*t*, **233**
 branching operations and, 479, 479*t*

DOU, *continued*

built-in functions and, 119
comparison and, 110–111, **110**, **111**
conditioning and branching operations using, 451*t*
END, 283, 283*t*, **283**
fixed-format in, 220, 220*t*
free-format syntax in, 221*t*
indicator use and, 444
ITER and, 206*t*, 306, 216*t*, 221*t*, 306*t*, **306**, 451*t*, 468*t*, 479, 479*t*
leave (LEAVE) and, 207*t*, 217*t*, 221*t*, 299, 309, 309*t*, **309**, 451*t*, 468*t*, 479, 479*t*
natural expression support in, 220, 220*t*
nesting in, 227
OR, 340, 340*t*, **340**
syntax diagram for, 216*t*

double-byte characters set (DBCS), convert to (%GRAPHIC), 158, **158**

DOW, 21, 105, 128, 206*t*, 275–277, 275*t*, **276**, **277**, 443, 450, **450**, 455, 467–469, **468**, 468*t*, **469**
AND, 233, 233*t*, **233**
branching operations and, 479, 479*t*
built-in functions and, 119
comparison and, 110–111, **110**, **111**
conditioning and branching operations using, 451*t*
END, 283, 283*t*, **283**
fixed-format in, 220, 220*t*
free-format syntax in, 221*t*
indicator use and, 444, 445
ITER and, 206*t*, 306, 216*t*, 221*t*, 306*t*, **306**, 451*t*, 468*t*, 479, 479*t*
leave (LEAVE) and, 207*t*, 217*t*, 221*t*, 299, 309, 309*t*, **309**, 451*t*, 468*t*, 479, 479*t*
natural expression support in, 220, 220*t*
nesting in, 227
OR, 340, 340*t*, **340**
READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
syntax diagram for, 216*t*

DOWNTD infix operator, 469
DOWNTD keyword, 298, 299
DSPLY, 26, 206*t*, 278–279, 278*t*, **279**
free-format syntax in, 221*t*
syntax diagram for, 216*t*

DTAARA keyword, 62*t*, 501, 502
*DTAARA, 28*t*, 305, 342, 398, 503

DEFINE and, 33*t*, 205*t*, 209, 215*t*, 263–265, 263*t*, **264**, **265**, 503–505, **504**, **505**
DTStoTM RPG routine , 639*t*
dumb terminals, 551. *See also* workstations
DUMP, 206*t*, 280, 280*t*, **280**
free-format syntax in, 221*t*
syntax diagram for, 216*t*
duration built-in functions, 148, **148**
duration codes, 146, 146*t*, 187, 211–212, 211*t*, 296
add duration (ADDUR), 231, 231*t*, **230**
subtract duration (SUBDUR), 208*t*, 211, 212, 218*t*, 385–386, 385*t*, **386**, 580*t*
dynamic call (CALL), 205*t*, 215*t*, 239–240, 239*t*, **239**, **240**, 343, 344, 345, 347, 433–434, 435, 444, 594–595, 594*t*, **595**, **595**

E

E operation extender, 212, 213*t*, 222, 224
EDI applications, 143
edit codes, 96–98, 97–98*t*
dates and, 99, 99–100*t*
edit with (%EDITC), 149–150, **150**
edit with floating point (%EDITFLT), 151
edit words, 92–94, 93*t*, **94**, **96**
control codes in, 94, 95*t*
edit with (%EDITW), 151, **151**
limits to, 10*t*
literal values and, 95
named constants and, 95, **96**
%EDIT, edit codes and, 96
%EDITC, 122*t*, 149–150, **150**
%EDITFLT, 122*t*, 151
editing numeric output, 92–99. *See also* decimal data errors, decimal notation, numeric data
date edit code controls for, 99, 99–100*t*
edit codes and, 96–98, 97–98*t*
edit word control codes in, 94, 95*t*
edit words in, 92–94, 93*t*, **94**
ending positions and, 93
named constants and, using edit words with, 95, **96**

editors for RPG, 4, 6
fixed- to free-format translation and, 584
%EDITW, 122*t*, 151, **151**
/EJECT directive, 30*t*
%ELEM, 120, 122*t*, 152, **152**
return values from, at compile time, 119
elements, number of, in arrays and tables (%ELEM), 152, **152**

- ELSE, 33*t*, 206*t*, 281, 281*t*, **281**, 282, 471
 conditioning and branching operations using, 451*t*
 conditioning indicators not supported by, 209
 free-format syntax in, 221*t*, 566
 SELECT, 373–374, 373*t*, **373**
 syntax diagram for, 216*t*
- ELSEIF, 5, **5**, 206*t*. *See also* IF
 conditioning and branching operations using, 451*t*
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
- ELSEIF DEFINED conditional preprocessor directive, 33*t*
- ELSEIF NOT DEFINED conditional preprocessor directive, 33*t*
- END, 28*t*, 206*t*, 271, 283, 283*t*, **283**
 conditioning and branching operations using, 451*t*
 free-format syntax and, 566
 syntax diagram for, 216*t*
- end marker, loops, 272
- /END-FREE directive, 31*t*, 112
- ENDCS, 206*t*
 compare and perform subroutine (CAS), 21, 205*t*, 215*t*,
 220, 220*t*, 234, 245–247, 245*t*, **245**, **246**, 283, 283*t*,
283, 444, 451*t*, 467*t*
 conditioning indicators not supported by, 209
 syntax diagram for, 216*t*
- ENDDO, 16, 206*t*, 271, 273, 274, 275, 309
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
- /END-EXEC directive, 31*t*
- ENDFOR, 206*t*, 298
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
- ENDIF, 5, 206*t*, 281, 282, 302, 373, 457, 471, 473. *See also* IF
 conditioning indicators not supported by, 209
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
- /ENDIF conditional preprocessor directive, 33*t*
- ending positions, edit words and, 93
- ENDMON (End Monitor Group), 206*t*, 314, 337, 338
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
- ENDSL (End Subroutine), 5, 206*t*, 373
 conditioning indicators not supported by, 209
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
- ENDSR (End Subroutine), 9, 11, 206*t*, 284–285, 284*t*, **285**,
 234, 301, 310, 341, 373, 478
 branching using, 479, 479*t*
 conditioning indicators not supported by, 209
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
 enhanced or alternate calculation specifications syntax, 563,
 564–564, **564**. *See also* free-format syntax
 *ENTRY 28*t*, 346
 entry parameter list (entry PLIST), 487
 entry points, 465, 487, 518–523
 environment variables, 592–593, 593*t*
 CONTENT_LENGTH as, 592, 593*t*, 595
 get (QtmhGetEnv) , 588–590, 589*t*, **589**, **590**
 HTTP_COOKIE as, 593*t*
 HTTP_USER_AGENT as, 592, 593*t*
 put (QtmhPutEnv) in, 590–592, 591*t*, **591**
 QUERY_STRING as, 592, 593*t*
 REQUEST_METHOD as, 592, 593*t*, 595
 EOF conditional preprocessor directive, 33*t*, 33
 %EOF, 122*t*, 153–154, 153*t*, **154**
 EQ, 220, 302, 340, 402, 470
 %EQUAL, 122*t*, 154–155, 154*t*, **155**, 311, 376, 377
 equi-join files, 540
 error handling
 blank transparency and, 622
 decimal data errors and, 619–629
 display message (DSPLY), 278–279, 278*t*, **279**
 %ERROR, 122*t*, 155, **155**, 222, 224
 file status error codes in, 183*t*, 224–225, **224**, 225–226*t*, 337*t*
 FIXNBR parameter for, 621–623, 629
 monitor group creation (MONITOR) for, 207*t*, 217*t*,
 221*t*, 313–314, 313*t*, **313**, **314**, 337–338
 program status error codes in, 184*t*, 222, **222**, 223*t*, 337*t*
 special device files and, 348
 test error status condition (ON-ERROR), 207*t*, 217*t*, 220,
 220*t*, 337–338, 314, 314*t*
 XML parser , %STATUS extended exception/error codes
 for, 125*t*, 155, 182–184, **182**, 183*t*, 222, 224,
 413–415*t*, 420–422*t*
 XML parser exception/error codes and, 165–166*t*
 error indicator, 16
 %ERROR, 122*t*, 155, **155**, 222, 224
 escape characters and escape sequences, Web (CGI) pro-
 gramming interfaces and, 601
 eServer iSeries, xxii
 *EUR , 141*t*, 189*t*, 210*t*, 211*t*, 318*t*, 327, 327*t*, 634*t*, 636*t*
 separator for, 635
 EVAL (Evaluate Two Items), 4, 105, 108–110, **109**, 206*t*,
 222, 286–287, 286*t*, **287**, 287*t*, 378, 379, 382, 388,
 570–571, 570*t*, **571**, 571*t*
 %ADDR (get address) and, 119, 120, 120*t*, 129–130, **130**
 assignment and, 108
 built-in functions and, 119

- EVAL, *continued*
 free-format syntax and, 221*t*, 569–570, **569**, **570**, 584
 natural expression support in, 220, 220*t*
 syntax diagram for, 216*t*
- EVALR (Evaluate, Right-Adjust), 105, 108–110, **109**, 116, 206*t*, 288–289, 288*t*, **289**, 570–571, 570*t*, **571**, 571*t*
 assignment and, 108
 built-in functions and, 119
 free-format syntax in, 221*t*
 natural expression support in, 220, 220*t*
- event codes, XML-SAX generated, 164*t*, 423–424*t*
- evolution of computer programming, xix
- EXCEPT (Exception), 9, 88, 206*t*, 250
 cursor position in file after execution of, 543*t*, **545**, **546**
 file processing using, 530*t*
 READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
 READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
 READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
 syntax diagram for, 216*t*
- /EXEC SQL directive, 31*t*
- executables, 433
- EXFMT (Extend Format), 206*t*, 293, 293*t*, **293**, 554–555, **554**
 file processing using, 530*t*
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
 workstation device files and, 552*t*
- EXIT, 21
- exit points, 465, 484
- exponentiation, 107, 108
- EXPORT keyword, 63*t*, 92*t*, 435, 437–439, **438**, **439**
- exporting procedures, 509–510, **510**
- expressions, 103–116
 assignment and, 105, 108–110, **109**, **111**
 Boolean, 108
 Calculation specification and, 112–113, **113**
 calling procedures using, 511, **511**
 %CHAR and, 121*t*, 137–138, **138**, 571–572, **571**, **572**, 582
 comparison and, 105, 110–111, **110**
 continuation of, 108
 convert to packed decimal (%DEC) and, , 121*t*, 142–144, **143**, **144**, **573–574**, 574, 588
 convert to packed decimal (%DECH) and, , 121*t*, 142–144, **143**, **144**
 declarations and, 105
 Definition specification and declarative statements using, 112, **112**
 evaluate, right-adjust (EVALR) and, 105, 108–110, **109**, 116, 119, 206*t*, 220, 220*t*, 221*t*, 288–289, 288*t*, **289**, 570–571, 570*t*, **571**, 571*t*
 evaluation (EVAL) and, 4, 105, 108–110, **109**, 119–120, 120*t*, 129–130, **130**, 206*t*, 216*t*, 220, 220*t*, 221*t*, 222, 286–287, 286*t*, **287**, 287*t*, 378, 379, 382, 388, 569–571, **569**, **570**, 570*t*, **571**, 571*t*, 584
 fixed-format code and, 106
 free-format code and, 111, 112–113, **113**
 infix notation and, 105
 l-value in, 108–110, **109**
 natural, 105–116, 220, 220*t*, 222
 numeric, 108
 operands and operators in, 106, 106*t*
 parameters and, 113, 114–115, **114**, **115**
 powers and roots syntax in, 108
 precision rules for, 115–116
 priority of operators in, 107–108, 107*t*
 procedures and, 105
 r-value in, 108–110, **109**
 return values and, 113
 string, 108
 tokens (symbols) in, 108, 204*t*
- EXPROPTS keyword, 41*t*, 116
- EXSR (Execute Subroutine), 206*t*, 234, 249–295, 294*t*, **295**, **296**
 free-format syntax in, 221*t*
 syntax diagram for, 216*t*
- EXTBININT keyword, 41*t*
- extended factor 2, 203
- extended name specification, **53**, 55–56, 56*t*, **56**
- extenders (E), 212, 213*t*
- external vs. internal format, dates and times, 635
- externally described files
 exception (EXCEPT), 9, 88, 206*t*, 216*t*, 250, 351–361, 530*t*, 543*t*, **545**
 file field descriptions in, 76, 77*t*, 90, 90*t*
 Input specifications and, 70, **70**, 76, 76*t*
 Output specification and, 88, 88–89*t*, 90*t*
- EXTFILE keyword, 48*t*, 49*t*, 339
- EXTFLD keyword, 63*t*
- EXTFMT keyword, 63*t*
- EXTIND keyword, 48*t*
- EXTMBR keyword, 49*t*, 339
- EXTNAME keyword, 63*t*, 502
- EXTPGM keyword, 64*t*
 prototyped parameters and, 515*t*

- EXTPROC keyword, 64*t*
 prototyped parameters and, 515*t*, 516*t*
- EXTRCT (Extract String), 206*t*, 211, 212
 data/time manipulation using, free- vs. fixed-format syntax, 580*t*
 syntax diagram for, 216*t*
- F**
- factor 1, 203
 factor 2, 203
- FEOD (Force End of Data), 206*t*, 250, 297, 297*t*, **297**
 cursor position in file after execution of, 543*t*, **546**
 file processing using, 530*t*
 free-format syntax in, 221*t*
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
 syntax diagram for, 216*t*
- field, programs, length of, 203
- field definition specification, **54**
- field indicators, 19
- field record-relation indicators, 19
- fields, 9, 531, 623
 define key list for (KLIST), 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308**, 530*t*
 field definition specification in, **54**
 get/set substring of a value in (%SUBST), 109, 120, 126*t*, 128, 187–189, **188–189**, 288, 387
 key field in (KFLD), 206*t*, 209, 217*t*, 307, 307*t*, **307**, 530*t*
 length of, 10*t*
 matched, 10*t*
 naming conventions for, 11*t*
 naming, 10*t*
 null-capable (%NULLIND), 172, **172**
 reserved names in, 26, 26*t*, 27, 27*t*
 result, 3
 set length of (%LEN), 119, 124*t*, 170, **170**
 target and source, 3
 test zone of (TESTZ), 208*t*, 218*t*, 395, 395*t*, **395**
 updating, identifying (%FIELDS), 156, **156**
 verify field content (%CHECKR), 140, **140**
- %FIELDS , 122*t*, 128, 156, **156**
- figurative constants, 21–27, 23–24*t*
 MOVEA and, 207*t*, 217*t*, 222, 322–323, **323**, 323*t*, 378, 379
 reserved words in, 27, 27*t*
- file. *See* files and file processing
- file objects, 532
- file processing. *See* files
- File specification, 4, 7–8, 7*t*, 37, 37*t*, 44–53, 44–53*t*
 built-in functions and, 120
 keywords for, 48–53*t*
 SQL and, 44
- file status indicator, 16
- *FILE, 314*t*, 337
- files and file processing, 9, 527–559. *See also* databases
 access path in, 531, 542, 549–552
 accessing, 542–548
 ACQ and, 205*t*, 221*t*, 215*t*, 228–229, 228*t*, **229**, 332, 364, 530*t*, 552*t*
 CHAIN and, 19, 157–158, 157*t*, **158**, 168–169, **169**, 205*t*, 215*t*, 221*t*, 224, 249–251, 249*t*, **250–251**, 268, 308, 308*t*, **308**, 351–361, 530*t*, 533, **533**, 553, **553**, 543*t*, **545**, 553*t*, 563, **563**, 623
 close (CLOSE), 205*t*, 215*t*, 221*t*, 250, 259, 259*t*, **259**, 297, 351–361, 552*t*, 553–554, **554**
 COMMIT and, 205*t*, 215*t*, 221*t*, 260–261, 260*t*, **261**, 370, 530*t*
 copy data (MOVE), 3, 4, 20, 207*t*, 209, 210, 212, 217*t*, 222, 316–319, 316*t*, 318*t*, **318**, **319**, 378, 379, 444, 568, **569**, 570–571, 570*t*, **571**, 571*t*, 580*t*
 copy data, left-justified (MOVE), 207*t*, 209, 212, 271*t*, 281, 324–328, 324*t*, **325**, **326**, 327*t*, **328**, 344, 568, **569**, 570–571, 570*t*, **571**, 571*t*
 creating, 532–537, **532**, 533*t*, **533**
 cross-reference in, 531
 cursor positioning in, after successful operation, 543*t*
 database, 531, 532. *See also* databases
 DDS to define, 532, **532**, 539, **539**, 540–541, **540**, **541**, 549, **549**, 551, **551**
 define (DEFINE), 33*t*, 205*t*, 209, 215*t*, 263–265, 263*t*, **264**, **265**, 503–505, **504**, **505**
 delete (DELETE), 88, 157–158, 157*t*, **158**, 168–169, **169**, 205*t*, 215*t*, 221*t*, 250, 266–268, 266*t*, **267**, **268**, 308, 308*t*, 355–361, 530*t*, 543*t*, **545**
 directories and subdirectories for, 430–431
 end of file (%EOF), , 122*t*, 153–154, 153*t*, **154**
 equi-join, 540
 exception (EXCEPT), 9, 88, 206*t*, 216*t*, 250, 351–361, 530*t*, 543*t*, **545**

Files and file processing, *continued*

- EXFMT and, 206*t*, 216*t*, 221*t*, 293, 293*t*, **293**, 530*t*, 552*t*, 554–555, **554**
 - fields in, 531
 - file objects and, 532
 - file status error codes in, 183*t*, 224–225, **224**, 225–226*t*, 337*t*
 - flat-, 429–431, 623
 - force end of data (FEOD), 206*t*, 216*t*, 221*t*, 250, 297, 297*t*, **297**, 351–361, 530*t*, 543*t*, **546**
 - force input priority (FORCE), 206*t*, 216*t*, 221*t*, 300, 300*t*, **300**
 - full-procedural, 338, 529–531, **529**
 - indexes in, 531, 549
 - Integrated File System (IFS) and, 429–431
 - join logical, 531, 540–542, **541**, **542**
 - key fields and key files (KFLD) in, 530*t*, 531, 533, 534–537, **534**, 534*t*, **535–536**, 539, 542, 549–551
 - key in, 531
 - key length of, 10*t*
 - key lists (KLIST) and, 530*t*, 550–551, **550**, **551**
 - libraries and, 531
 - logical, 531, 538–542
 - members in, 531
 - naming conventions in, 11, 11*t*
 - open (OPEN), 207*t*, 217*t*, 221*t*, 338–339, 338*t*, 339*t*, **339**, 356, 530*t*, 543*t*, **546**, 552*t*, 553–554, **554**
 - limit to number of, 10*t*
 - open status check (%OPEN), 174, **174**
 - operation codes used with, 530*t*, 543*t*
 - physical, 531, 532–537
 - post device information (POST) and, 207*t*, 217*t*, 221*t*, 349–350, 349*t*, **350**, 530*t*, 552*t*, 555, **555**
 - primary, 10*t*
 - printer, 10*t*
 - processing of, 527–559
 - product structure file example of, 536–537, 537*t*, **537**
 - random access and, 531, 533, 553, **553**
 - read from (READ), 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
 - read next changed record in (READC), 207*t*, 217*t*, 221*t*, 354, 354*t*, **354**, 530*t*, 552*t*, 557, **557**
 - read next record with equal key in (READE), 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
 - read prior record from data file in (READP), 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
 - read prior record with equal key in (READPE), 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
 - record-address in, 10*t*
 - records in, 531
 - REL and, 207*t*, 218*t*, 221*t*, 292, 530*t*, 552*t*
 - relational database, 623
 - rollback change in (ROLBK), 208*t*, 218*t*, 221*t*, 370, 370*t*, **370**, 530*t*
 - select/omit operations and, 539–540, 540*t*
 - sequential, 543
 - set greater than (SETGT), 9, 123*t*, 128, 168–169, **169**, 157–158, 157*t*, **158**, 206, 208*t*, 209, 218*t*, 221*t*, 250, 374–375, 374*t*, **357**, 530*t*, 543*t*, **548**
 - set lower limit (SETLL), 123*t*, 128, 168–169, **169**, 208*t*, 218*t*, 221*t*, 250, 376–377, 376*t*, **377**, 530*t*, 543*t*, **548**
 - source files in, 429–431
 - status code for (%STATUS), 125*t*, 155, 182–184, **182**, 183*t*, 222, 224, 413–415*t*, 420–422*t*
 - subsets of, processing using logical files, 538
 - unlock (UNLOCK), 208*t*, 218*t*, 221*t*, 250, 263, 292, 398–399, 398*t*, **399**, 501, 503–505, **504**, **505**, 530*t*, 543*t*
 - update (UPDATE), 88, 156, **156**, 208*t*, 219*t*, 221*t*, 250, 400–402, 400*t*, **401**, **402**, 530*t*, 543*t*, **548**, 552*t*/557–558, **558**
 - views in, 531
 - workstation device file processing and, 551–552, 552*t*
 - workstation format, write/read (EXFMT), 206*t*, 216*t*, 221*t*, 293, 293*t*, **293**, 530*t*, 552*t*, 554–555, **554**
 - write to (WRITE), 88, 208*t*, 219*t*, 221*t*, 404–405, 404*t*, **405**, 529, 530*t*, 543*t*, **548**, 552*t*, 554–555, **554**, 558, **559**
- first-page indicators, 19
 - fixed-format syntax, 3, 563, 564, **564**
 - expressions and, 106
 - free-format translation and, 584
 - fixed separators, date and time, 635
 - FIXNBR parameter, 42*t*, 616–617
 - decimal data errors and, 621–623, 629
 - flag setting, in structured programming, 465
 - flat-file systems, 429–431, 623
 - %FLOAT, 123*t*, 157, **157**, 167
 - floating-point data type, 12*t*
 - convert to (%FLOAT), 123*t*, 157, **157**, 167
 - edit with (%EDITFLT), 151
 - FLTDIV, 42*t*
 - FOR, 105, 128, 206*t*, 298–299, 298*t*, **299**, 467–469, **468**, 468*t*, **469**
 - built-in functions and, 119
 - comparison and, 110–111, **110**, **111**

- free-format syntax in, 221*t*
- indicator use and, 444
- infix operators for (by, to, downto), 468–469
- %INT and, 123*t*, 167–168, **168**, 588
- ITER and, 206*t*, 306, 216*t*, 221*t*, 306*t*, **306**, 451*t*, 468*t*, 479, 479*t*
- leave (LEAVE) and, 207*t*, 217*t*, 221*t*, 299, 309, 309*t*, **309**, 451*t*, 468*t*, 479, 479*t*
- natural expression support in, 220, 220*t*
- nested, 10*t*, , 227
- syntax diagram for, 216*t*
- FORCE (Force Data Input), 206*t*, 300, 300*t*, **300**
 - free-format syntax in, 221*t*
 - syntax diagram for, 216*t*
- format codes
 - for dates, 318, 327*t*, 634*t*
 - for time, 189*t*, 327*t*, 636*t*
 - for timestamps, 190*t*, 636–637
- formats, naming conventions for, 11*t*
- FORMLEN keyword, 49*t*
- FORMOFL keyword, 18, 49*t*
- FORMSALIGN, 42*t*
- FORTRAN, xix, xx, xxii
- %FOUND, 123*t*, 157–158, 157*t*, **158**, 371–372, 371*t*, **372**, 376, 377
 - SCAN and, 16, 157–158, 157*t*, **158**, 208*t*, 218*t*, 218, 371–372, 371*t*, **372**
- free-format syntax, xxiii, 3, 4, **4**, 6, 561–584
 - arrays and, 574–577, **575**, 576*t*, **577**, 577*t*
 - Calculation specifications using, 565–570
 - CALLP and, , 105, 113, 119, 205*t*, 220, 220*t*, 221*t*, 243, 243*t*, **244**, 429, 435, 584, 587, 588–592, 589*t*, **589**, **590**, 591*t*, **591**. *See also* prototyped calls/procedures
 - CHAIN example of, 563, **563**
 - comments in, 565–566, **565**
 - data structures and, 574–577, **575**, 576*t*, **577**, 577*t*
 - data type conversion in, 571–577
 - date and time processing using, 579–582, **580**, **581**
 - date value retrieval at runtime using, 582–583, **582**, **583**
 - DIM keyword and, 576, 576*t*
 - directives in, 566, **566**
 - /END-FREE directive in, , 31*t*, 112, 563, 584
 - EVAL and EVALR using, , 570–571, 570*t*, **571**, 571*t*, 584
 - expressions and, 111, 112–113, **113**
 - /FREE-FORM directive in, , 31*t*, 112, 563, 584
 - math operations using, 569–570, **569**, **570**
 - MOVE and MOVEL using, 568, **569**, 570–571, 570*t*, **571**, 571*t*
 - nested, 227, **227**
 - operation code syntax using, 566–569, **567**, **569**
 - parentheses use in, 584
 - procedures, subprocedures, subroutines and, declaring in, 577–584, **577**, **578**, **579**
 - programs, 220–221, 220*t*, 221*t*
 - QUALIFIED keyword and, 576, 576*t*
 - semicolon statement-terminator in, 565, 567, 584
 - SQL and, 584
 - translating fixed-format to, 584
 - /FREE and /END-FREE, 31*t*, 112, 563, 584
 - FROMFILE keyword, 64*t*
 - FTRANS, 42*t*
 - full procedural files, 338
 - function key attention identification bytes, 17–18, 17*t*, **18**, , **650**
 - function keys, indicators and, 16
 - function modules, 474, 484
 - functions, built-in. *See* built-in functions

G

 - GE, 220, 302, 340, 402, 470
 - general-purpose procedures, 510
 - GENLVL parameter and keywords, 42*t*
 - Create Bound Program (CRTBNDRPG) and, 607
 - GetDate RPG routine , 640*t*
 - GetDateAsString RPG routine, 641*t*
 - GetDayCount RPG routine , 640*t*
 - GetDayEx RPG routine , 639*t*
 - GetDayName RPG routine , 639*t*
 - GetDayOfWeek RPG routine , 639*t*
 - GetDayOfYear RPG routine , 640*t*
 - GetEndOfMonth RPG routine , 639*t*
 - *GETIN, 29*t*
 - GetProcAddr, 175
 - GetWeek RPG routine , 640*t*
 - GetWeekISO RPG routine , 640*t*
 - global variables, 431
 - EXPORT keyword and, 437–439, **438** , **439**
 - IMPORT keyword and, 437–439, **438** , **439**
 - scope and, 435–436, **436**
 - GOTO, 9, 206*t*, 301, 301*t*, **301**
 - branching using, 479, 479*t*
 - conditioning and branching operations using, 451*t*
 - ENDSR and, 9, 11, 206*t*, 209, 216*t*, 221*t*, 284–285, 284*t*, **285**, 234, 301, 310, 341, 373, 478, 479, 479*t*
 - EXSR and, 206*t*, 216*t*, 221*t*, 234, 249–295, 294*t*, **295**, **296**
 - indicator use and, 447
 - syntax diagram for, 216*t*

GOTO, *continued*

TAG and, 9, 11, 208*t*, 209, 218*t*, 389, 389*t*, **389**, 447, 478, 479, 479*t*

%GRAPH, 123*t*

graphic (DBCS) data type, 12*t*

%GRAPHIC, 158, **158**

graphics

convert to double-byte value (%GRAPHIC), 158, **158**

greater than (SETGT) , 9, 123*t*, 128, 168–169, **169**,

157–158, 157*t*, **158**, 206, 208*t*, 209, 218*t*, 221*t*, 250, 374–375, 374*t*, **357**, 530*t*, 543*t*, **548**

greater than operator, 106

GT, 220, 302, 340, 402, 470

H

H operation extender, 213*t*

H1–H9 indicators, 14*t*

handler subprocedure, XML parser and, 409–411, 415–418, 422–424, **422**

%HANDLER, 123*t*, 128, 159–161, **160**, **161**, 162–166, **163**, 165*t*, 409, 410, 415–418, **416**

Header specification, 4, 8*t*, 37, 37*t*, 38–44, 38–44*t*

keyword summary for, 38–44*t*

HELP key, 21

hexadecimal literal values, 22–27, 23*t*

hierarchy chart, 484–485, **485**

application, 485

program, 485

subprocedures, 457

high-level programming languages, 3, 9

*HIVAL*t*, 23*t*, 637–638, 638*t*

*HMS , 189*t*, 211*t*, 636*t*

HOME key, 21

host variables, data areas and, 503–505, **504**

*HOURS , 146*t*, 187, 211*t*, 231, 296

%HOURS, 123*t*, 148

HTTP_COOKIE , 593*t*

HTTP_USER_AGENT, 592, 593*t*

I

IF, xxi, 5, **5**, 21, 105, 128, 206*t*, 302–304, 302*t*, **303**, **304**, 455, 457, 467, **467**, 467*t*

AND and, 233, 233*t*, **233**, 472–473, **473**

built-in functions and, 119

compare and perform subroutine (CAS), 21, 205*t*, 215*t*, 220, 220*t*, 234, 245–247, 245*t*, **245**, **246**, 283, 283*t*, **283**, 444, 451*t*, 467*t*

comparison and, 110–111, **110**, **111**

conditioning and branching operations using, 451*t*

DO loops and, 448–450, **448**, **449**, **450**

ELSE and, 281, 281*t*, **281**

ELSEIF, 282, 282*t*, **282**

END, 283, 283*t*, **283**

free-format syntax in, 221*t*, 566

IF-THEN-ELSE and, 470–474. *See also* IF-THEN-ELSE

indicator use and, 444, 447

natural expression support in, 220, 220*t*

nested, 10*t*, 473, **473**

OR and, 340, 340*t*, **340**, 472–473, **473**

relationship testing using, 472–474, **472**, **473**

SELECT and, 373–374, 373*t*, **373**

syntax diagram for, 216*t*

IF DEFINED conditional preprocessor directive, 33*t*

IF NOT DEFINED conditional preprocessor directive, 33*t*

IF-THEN-DO, 5

IF-THEN-ELSE, xxi, 304, **340**, 443, 465, 467, 470–474

Boolean operators for, 470, 470*t*

nesting in, 227

IF-THEN-ELSE-END

indicator use and, 445

IFEQ, built-in functions and, 119

IFLE, 281

IGNORE keyword, 49*t*

ILE. *See* integrated language environment

IMPORT keyword, 64*t*, 437–439, **438** , **439**

IN (Read In a Data Area), 206*t*, 263, 305, 305*t*, **305**

data areas and, 501, 503–505, **504**, **505**

free-format syntax in, 221*t*

syntax diagram for, 216*t*

*IN, 20

in-line CASE structures, 5, 474–476, **474**

nesting , 403, **403**

OTHER and, 341, 341*t*, **341**

start (SELECT) and, 373–374, 373*t*, **373**

WHEN and, 402–403, 402*t*, **403**

in-line ILE procedures, 648, **648**, **649**

INCLUDE keyword, 49*t*

/INCLUDE directive, 30*t*, 30

INDDS keyword, 49*t*

INDENT parameter and keywords, 42*t*

Create Bound Program (CRTBNDRPG) and, 613

index and increment value, in loops, 298

indexes, file, 531, 549

- indicator data type, 12*t*
- indicators , 13–16, **13**, 14–15*t*, 443–446, 455
- arrays for, 20
 - avoid use of, 20–21
 - classification of, 18–20
 - conditioning, 19, 21, 203 , 209, 209*t*
 - control-level, 18–19, 203
 - controlling, 19
 - default function key, 21
 - device files and, 16
 - error, 16
 - field, 19
 - field record-relation, 19
 - file status, 16
 - first-page, 19
 - %FOUND versus, 157–158, 157*t*, **158**
 - function keys and, 16
 - indicatorless code vs., 14, **14**, 20–21, 447–448, **447**
 - last-record, 20
 - level-break, 18–19, 20
 - logic and, 445–446, **445**, **446**
 - matching-record, 20–21
 - named, 20
 - overflow, 18
 - program-termination, 16
 - record-identifying, 18
 - resulting, 19
 - set off (SETOFF), 208*t*, 218*t*, 378, 378*t*, **378**
 - set on (SETON), 20, 208*t*, 218*t*, 379, 379*t*, **379**
 - testing for ranges, lists, select/omits using, 453
 - trash, 16
 - usage of, 15–16
 - variables for, 20
- INFDS (Input/Output Feedback Data Structure), 50*t*, 59, 59–60*t*
- infix notation, expressions and, 105
- infix operators, FOR and (by, to, downto), 468–469
- INFSR, 50*t*, 294
- Input specification, 4, 7–8, 7*t*, 37, 37*t*, 69–77
- externally described files and, 70, **70** , , 76, 76*t*
 - program described files and, 69–75, **70** , 71–73*t*, 73–75*t*
 - record identifying entries and, 69
- input, RPG, 9
- decimal data errors and, 628–629
 - force next (NEXT), 207*t*, 217*t*, 221*t*, 332–333, 332*t*, **333**
- %INT, 123*t*, 167–168, **168**, 588
- integer data types, 12*t*, 56–58, 57*t*, 58*t*
- convert to (%INT), 123*t*, 167–168, **168**, 588
 - convert to unsigned (%UNS), 194–195, **195**
 - convert to unsigned (%UNSH), 194–195, **195**
 - Definition specification and, 56–58, 57*t*, 58*t*
 - %DIV, 147, **147**
 - limits/capacities of, 57–58, 57*t*, 58*t*
- integrated development environments (IDE), 6
- Integrated File System (IFS) , 429–431
- integrated language environment (ILE), xxiii, xxiv, 2, 607
- in-line procedures for, 648, **648**, **649**
- %INTH, 123*t*
- INTPREC keyword, 42*t*
- INZ keyword, 25, 65*t*, 366, 367
- *INZSR, 28*t*
- %INZSR, 334
- iSeries, xxiv
- *ISO, 141*t*, 189*t*, 190*t*, 210*t*, 211, 211*t*, 318*t* 327, 327*t*, 634*t*, 636*t*
- separator for, 635
- *ISO0, 190*t*
- ITER (Iterate Loop), 206*t*, 306, 306*t*, **306**, 468*t*
- branching using, 479, 479*t*
 - conditioning and branching operations using, 451*t*
 - free-format syntax in, 221*t*
 - syntax diagram for, 216*t*
- iteration of loops, 298
- ITER and, 206*t*, 306, 216*t*, 221*t*, 306*t*, **306**, 451*t*, 468*t*, 479, 479*t*
 - structured programming and, 466, 467–469, **468**, 468*t*, **469**
- ## J
- Java, integer data type capacity limit in, 57
- JFLD keyword, 541
- *JIS, 141*t*, 189*t*, 210*t*, 211*t*, 318*t*, 327, 327*t*, 634*t*, 636*t*
- separator for, 635
- *JOB, 210*t*, 211*t*, 634*t*, 636*t*
- *JOBRUN, 210*t*, 211*t*, 318*t*
- jobs, end of (%SHTDN), 179
- JOIN keyword, 541
- join logical file, 531, 540–542, **541**, **542**
- equi-join, 540
- JREF keyword, 541–542
- *JUL, 141*t*, 210*t*, 318*t*, 327, 327*t*, 634*t*
- separator for, 635
- ## K
- KA–KY indicators, 14*t*, 20, 21
- %KDS , 123*t*, 128, 168–169, **169**

- key fields and key lists, 531, 533, 534, 542, 549–551, **550**, **551**
 - logical file with, 539
 - physical file with, 534–537, **534**, 534*t*, **535–536**
 - data structure to (%KDS), 123*t*, 128, 168–169, **169**
 - access path processing and, 549–552, **550**, **551**
 - define (KLIST), 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308**, 530*t*
 - key field in (KFLD), 206*t*, 209, 217*t*, 307, 307*t*, **307**, 530*t*
 - partial, 535–536, **535–536**
- keyed files
 - read next with equal key (READE), 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
 - read prior record with equal key (READPE), 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- KEYLOC, 50*t*
- keys, 531
- keywords, 3
 - Create Bound Program (CRTBNDRPG) and, 605
 - Definition specification, 61–69*t*
 - File specification, 48–53*t*
 - Header specification, 38–44*t*
 - Procedure specification, 92, 92*t*
 - prototyped calls/procedures, 500
- KFLD (Key Field), 206*t*, 307, 307*t*, **307**
 - conditioning indicators not supported by, 209
 - file processing using, 530*t*
 - syntax diagram for, 217*t*
- KLIST (Key List), 9, 206*t*, 307, 308, 308*t*, **308**
 - conditioning indicators not supported by, 209
 - file processing using, 530*t*
 - syntax diagram for, 217*t*
- L**
- l-value , 108–110, **109**, 288
- L0 indicator, 15*t*
- L1–L9 indicators, 15*t*, 18, 20
- labels, 9
 - define (TAG), 9, 11, 208*t*, 209, 218*t*, 389, 389*t*, **389**, 447, 478, 479, 479*t*
 - naming conventions for, 11*t*
- LANGID parameter and keywords
 - Create Bound Program (CRTBNDRPG) and, 614
- last record indicators, 20
- *LDA , 28*t*
- LE, 220, 302, 340, 402, 470
- LEAVE (Leave/Exit Loop), 207*t*, 299, 309, 309*t*, **309**, 468*t*
 - branching using, 479, 479*t*
 - conditioning and branching operations using, 451*t*
 - free-format syntax in, 221*t*
 - syntax diagram for, 217*t*
- LEAVESR (Leave/Exit Subroutine), 207*t*, 294, 310, 310*t*, **310**
 - branching using, 479, 479*t*
 - free-format syntax in, 221*t*
 - syntax diagram for, 217*t*
- left-signed data type, 12*t*
- %LEN, 119, 124*t*, 170, **170**
 - return values from, at compile time, 119
- less-than operator, 106
- level-break indicators, 18–19, 20, 295
 - DO and, 449–450, **449**
- libraries
 - files and file processing, 531
 - qualified program names and, 240
- LIKDES keyword, %HANDLER and, 160–161, **160**, **161**
- *LIKE, 28*t*
 - DEFINE and, 263–265, 263*t*, **264**, **265**
- LIKE keyword, 65*t*
 - DEFINE and, 263–265, 263*t*, **264**, **265**
- LIKEDS keyword, 11, 66*t*
 - DEFINE and, 263–265, 263*t*, **264**, **265**
- LIKEREC keyword, 66*t*
 - prototyped parameters and, 516*t*
- limitations of RPG, 9, 10*t*
- lines per page, 10*t*
- Linux, 429
- lists, 453–454, **453**, **454**. *See also* arrays
- literal values, 21–27, 23*t*
 - edit words using, 95
- local variables, 431
 - scope of, 435–436, **436**
- *LOCK, 28*t*, 305
- locked records
 - exception (EXCEPT), 9, 88, 206*t*, 216*t*, 250, 351–361, 530*t*, 543*t*, **545**
 - *LOCK, 28*t*, 305
 - READE and READPE and, record-lock contention
 - caused by, 544
 - record-lock contention caused by, 544
 - unlock (UNLOCK), 208*t*, 218*t*, 221*t*, 250, 263, 292, 398–399, 398*t*, **399**, 501, 503–505, **504**, **505**, 530*t*, 543*t*
- logic
 - IF-THEN-ELSE in, 470–474
 - indicator use and , 445–446, **445**, **446**
 - procedures, subprocedures, subroutines and, 510
 - structured programming and, 465, 484
 - top-down, 455

- logic control modules, 474, 484
 logic cycle, program, 431
 logic flow, in CASE structures, 476–478, **477**
 logical files, 531, 538–542
 DDS to define, 539, **539**
 join, 540–542, **541, 542**
 physical file mapped to, 538–542, **538**
 select/omit operations and, 539–540, 540*t*
 logical operators, 106
 *LONGJUL, 141*t*, 210*t*, 318*t*, 327, 327*t*
 LOOKUP (Lookup/Search), 16, 207*t*, 311–312, 311*t*, **312**
 %EQUAL and, 122*t*, 154–155, 154*t*, **155**, 311, 376, 377
 %FOUND versus indicators in, 157–158, 157*t*, **158**
 indicator use and, 444
 syntax diagram for, 217*t*
 %LOOKUP, 124*t*, 171, **171**
 %FOUND versus indicators in, 157–158, 157*t*, **158**
 looping
 branching operations and, 479, 479*t*
 do until (DOU), 273–274, 273*t*, **274, 275**
 do while (DOW), 275–277, 275*t*, **276, 277**
 DO, 271–273, 271*t*, **271, 272, 273**, 448–450, **448, 449, 450**
 ELSE, 281, 281*t*, **281**
 ELSEIF, 282, 282*t*, **282**
 end marker for, 272
 END, 283, 283*t*, **283**
 FOR, 298–299, 298*t*, **299**
 index and increment value in, 298
 infinite, 465, 484
 iterate (ITER) and, 206*t*, 306, 216*t*, 221*t*, 306*t*, **306**, 451*t*,
 468*t*, 479, 479*t*
 iteration of, 298, 466, 467–469, **468, 468*t*, 469**
 leave (LEAVE) and, 207*t*, 217*t*, 221*t*, 299, 309, 309*t*,
 309, 451*t*, 468*t*, 479, 479*t*
 limit to, 298
 OR and, 340, 340*t*, **340**
 start values in, 298
 structured programming and, 465, 466, 467–469, **468**,
 468*t*, **469**, 484
 WHEN and, 402–403, 402*t*, **403**
 *LOVAL, 23*t*, 637–638, 638*t*
 LR indicator, 15*t*, 16
 LT, 220, 302, 340, 402, 470
 machine-readable object or executables, 433
 mainline calculations, in subroutines, 295, **295**
 mainline procedures, 429, 431, **431**, 518
 mapping, physical to logical file, 538–542, **538**
 MAPVAL keyword, 638–639
 masks, 92. *See also* edit words
 matched fields, limits to, 10*t*
 matching record indicators, 20–21
 math operation codes, 456, **456**
 free-format syntax and, 569–570, **569, 570**
 MAXDEV keyword, 50*t*, 332
 *MDY *t*, 141*t*, 210*t*, 318*t*, 327, 327*t*, 634*t*
 separator for, 635
 members, file, 531
 memory (storage)
 allocate (ALLOC), 205*t*, 215*t*, 232, 232*t*, **232**, 362
 automatic vs. static, 436–447
 reallocate (REALLOC), 207*t*, 218*t*, 362–363, 362*t*, **363**
 release allocated (DEALLOC), 205*t*, 215*t*, 221*t*, 262,
 262*t*, **262**
 size in bytes used (%SIZE), 119, 120, 125*t*, 180–181,
 180*t*, **181**
 messages, display (DSPLY), 278–279, 278*t*, **279**
 METHOD-GET/METHOD-POST, 600–601
 MHHZO, 207*t*, 315, 315*t*
 syntax diagram for, 217*t*
 MHLZO, 207*t*, 315, 315*t*
 syntax diagram for, 217*t*, 217
 Microsoft Windows, 6, 429
 minus sign (minus operator), 106
 *MINUTES, 146*t*, 187, 211*t*, 231, 296
 %MINUTES, 124*t*, 148
 MLHZO, 207*t*, 315, 315*t*
 syntax diagram for, 217*t*
 MLLZO, 207*t*, 315, 315*t*
 syntax diagram for, 217*t*
 Modula-2, 443
 modules and modular programming, 431–435, 481–505,
 510. *See also* structured programming
 benefits of, 483–485
 calling in, 484
 Create RPG Module (CRTRPGMOD) and, 430, 434,
 510, 605
 data areas in, 501–505, **501**
 exit points in, 484
 function modules in, 474, 484
 hierarchy chart in, 484–485, **485**
 inter-module communication and, 481–505
 logic control in, 484

M

M operation extender, 213*t*
 M1–M9 indicators, 15*t*, 20

- modules and modular programming, *continued*
 logic control modules in, 474, 484
 looping in, 484. *See also* looping
 parameter passing in, 486–494. *See also* parameter passing
 PLIST and parameter passing in, 9, 207*t*, 209, 345,
 346–348, 346*t*, **347**, 348*t*, 487–493, **493**
 procedures, subprocedures, subroutines and, 509–510, **509**
 prototyped calls and, 494–500, **495**, **496**, **498**, 494
 redundant code and, 484
 reusable code in, 484, 485
 top-down design and, 484
- MONITOR (Create Monitor Group), 207*t*, 313–314, 313*t*,
313, **314**, 338
 free-format syntax in, 221*t*
 syntax diagram for, 217*t*
 test error status condition (ON-ERROR), 207*t*, 217*t*, 220,
 220*t*, 337–338, 314, 314*t*
- *MONTH, 26*t*, 27*t*, 146*t*, 187, 211*t*, 231, 296
- %MONTHS, 124*t*, 148
- MOVE (Move Data), 3, 4, 20, 207*t*, 209, 210, 212, 222,
 316–319, 316*t*, 318*t*, **318**, **319**, 378, 379
 data/time manipulation using, free- vs. fixed-format syn-
 tax, 580*t*
 free-format syntax and, 568, **569**, 570–571, 570*t*, **571**, 571*t*
 indicator use and, 444
 syntax diagram for, 217*t*
- MOVEA (Move Array), 207*t*, 222, 378, 379
 figurative constants and, 322–323, **323**, 323*t*
 syntax diagram for, 217*t*
- MOVEL (Move Data Left-Adjust), 207*t*, 209, 212, 281,
 324–328, 324*t*, **325**, **326**, 327*t*, **328**, 344
 date and time variables in, 326–328, 327*t*, **328**
 free-format syntax and, , 570–571, 570*t*, **571**, 571*t*
 free-format syntax and, 568, **569**
 syntax diagram for, 217*t*
- MR indicator, 15*t*, 20
- *MSECONDS, 146*t*, 187, 211*t*, 231, 296
- %MSECONDS, 124*t*, 148
- MULT (Multiply), 16, 207*t*, 222, 329–330, 329*t*, **330**, 456
 free-format syntax and, 569–570, **569**, **570**
 indicator use and, 444
 syntax diagram for, 217*t*
- multiple-occurrence data structures, 574–577, **575**, 576*t*,
577, 577*t*
- multiplication, 106
 multiply (MULT), 16, 207*t*, 217*t*, 222, 329–330, 329*t*,
330, 444, 456, 569–570, **569**, **570**
- MVR (Move Remainder), 207*t*, 331, 331*t*, **331**
 syntax diagram for, 217*t*
- N**
- N operation extender, 213*t*, 250, 297
- named activation groups, 609
- named constants, 9, 21–27, **25**
 limits to, 10*t*
 naming conventions for, 11*t*
 XML parser, 419
- named indicators, 20
- named parameter lists, 490–493, **490**
- named/unnamed parameters, 346–347
- naming conventions, 11, 11*t*
 reserved names in, 26, 26*t*
 reserved words in, 27, 27*t*
 structured programming and, 463
- natural expressions, in programs, 105–116, 220, 220*t*. *See also* expressions
- NE, 220, 302, 340, 402, 470
- nested code, 227, **227**
 CASE structures and, 474
 free-format, 227, **227**
 IF, 473, **473**
 in-line case operations, 403, **403**
 limits to, 10*t*
- NEXT (Get Next Record), 207*t*, 332–333, 332*t*, **333**
 free-format syntax in, 221*t*
 syntax diagram for, 217*t*
- *NEXT, 29*t*
- nibbles, 392
- NOMAIN keyword, 42*t*, 431, 435, **552–523**
 procedures, subprocedures, subroutines and, 509
- nonprogrammable terminals (NPT), 551. *See also* workstations
- NOOP keyword, 66*t*
 prototyped parameters and, 516*t*
- NOT, 106
 %BITNOT, 133–134, **133**, 134*t*
- *NULL, 23*t*
- null-capable fields, retrieve property of (%NULLIND),
 124*t*, 172, **172**
- %NULLIND, 124*t*, 172, **172**
- numeric data
 add duration to (ADDDUR), 231, 231*t*, **230**
 addition (ADD), 13, **13**, 14, **14**, 205*t*, 215*t*, 222, 230,
 230*t*, **230**, 444, 456, 569–570, **569**, **570**
 binary and zoned fields in, 622
 blank transparency and, 622
 clear (CLEAR), 205*t*, 215*t*, 221*t*, 256–258, 256*t*, 257*t*,
258, 317, 366, 425, 626

- convert to character (%CHAR), 121*t*, 137–138, **138**, 571–572, **571**, **572**, 582
- convert to floating point (%FLOAT), 123*t*, 157, **157**, 167
- convert to integer (%INT), 123*t*, 167–168, **168**, 588
- convert to packed decimal (%DEC), , 121*t*, 142–144, **143**, **144**, **573–574**, 574, 588
- convert to packed decimal (%DECH), 121*t*, 142–144, **143**, **144**
- convert to time value (%TIME), 126*t*, 189, 189*t*, 572–573, **573**, 581, 582–583, **582**, **583**
- convert to timestamp (%TIMESTAMP), 126*t*, 190, 190*t*, 581, 582–583, **582**, **583**
- convert to unsigned integer (%UNS), 194–195, **195**
- convert to unsigned integer (%UNSH), 194–195, **195**
- cross-foot array (XFOOT), 195, **195**, 208*t*, 219*t*, 406, 406*t*, **406**
- dates from, duration built-in functions for, 148, **148**
- decimal data errors and, 619–629
- division (%DIV), 147, **147**
- division (DIVIDE), 16, 205*t*, 222, 269–270, 269*t*, **270**, 456
- edit (%EDITC), 149–150, **150**
- edit (%EDITW), 151, **151**
- edit codes and, 97–98, 97–98*t*
- edit with floating point (%EDITFLT), 151
- editing of, 92–94, 93*t*, **94**. *See also* editing numeric output
- evaluate, right-adjust (EVALR), 105, 108–110, **109**, 116, 119, 206*t*, 220, 220*t*, 221*t*, 288–289, 288*t*, **289**, 570–571, 570*t*, **571**, 571*t*
- evaluation (EVAL), 4, 105, 108–110, **109**, 119–120, 120*t*, 129–130, **130**, 206*t*, 216*t*, 220, 220*t*, 221*t*, 222, 286–287, 286*t*, **287**, 287*t*, 378, 379, 382, 388, 569–571, **569**, **570**, 570*t*, **571**, 571*t*, 584
- FIXNBR parameter for, 621–623, 629
- math operation codes and, 456, **456**
- move remainder of division (MVR), 207*t*, 217*t*, 331, 331*t*, **331**
- multiply (MULT), 16, 207*t*, 217*t*, 222, 329–330, 329*t*, **330**, 444, 456, 569–570, **569**, **570**
- packed decimal data as, 620, 626–627, 627*t*, 628*t*
- remainder division (%REM), 177, **177**
- retrieve decimal positions (%DECPOS), 119, 121*t*, 145, **145**
- set greater than (SETGT) , 9, 123*t*, 128, 168–169, **169**, 157–158, 157*t*, **158**, 206, 208*t*, 209, 218*t*, 221*t*, 250, 374–375, 374*t*, **357**, 530*t*, 543*t*, **548**
- sign or status in, 624
- square root (SQRT), 208*t*, 218*t*, 222, 382–383, 382*t*, **383**, 444, 569–570, **569**, **570**
- subtract (SUB), 208*t*, 218*t*, 222, 383–384, 383*t*, **384**, 456, 569–570, **569**, **570**
- subtract duration (SUBDUR), 208*t*, 211, 212, 218*t*, 385–386, 385*t*, **386**, 580*t*
- test character field for (TESTN), 208*t*, 218*t*, 394, 394*t*, **394**
- test result field of (TEST), 16, 208*t*, 209, 218*t*, 221*t*, 390–391, 391*t*, **391**, 580*t*
- zero and add (Z-ADD), 26, 208*t*, 219*t*, 344 , 425, 425*t*, **425**
- zero and subtract (Z-SUB), 208*t*, 219*t*, 426, 426*t*, **426**
- zoned decimal data as, 620, 624–626, **624**, 625*t*
- numeric expressions, 108
- numeric literal values, 22–27, 23*t*. *See also* numeric data, above

O

- OA–OG indicators, 15*t*
- object-based programming, 461
- OCCUR, 207*t*
- syntax diagram for, 217*t*
- %OCCUR, 124*t*, 173, **173**
- occurrence index, 334
- OCCURS keyword, 25, 67*t*
- OF indicator, 20
- *OFF, 23*t*
- *OFL, 29*t*
- OFLIND keyword, 18, 50*t*
- *OMIT, 23*t*, 539–540
- *ON, 23*t*
- ON-ERROR (Test Error Status), 207*t*, 337–338
- figurative constants tested by, 314, 314*t*
 - free-format syntax in, 221*t*
 - monitor group creation (MONITOR), 207*t*, 217*t*, 221*t*, 313–314, 313*t*, **313**, **314**, 337–338
 - natural expression support in, 220, 220*t*
 - syntax diagram for, 217*t*
- opcode. *See* operation code
- OPDESC (operand description) keyword, 67*t*, 648
- prototyped parameters and, 516*t*
 - parameter passing and, 523–524
- OPEN (Open File), 207*t*, 338–339, 338*t*, 339*t*, **339**, 356, 553–554, **554**
- cursor position in file after execution of, 543*t*, **546**
 - file processing using, 530*t*
 - free-format syntax in, 221*t*
 - syntax diagram for, 217*t*
 - workstation device files and, 552*t*
- Open Query File (OPNQRYF), 21
- %OPEN, 124*t*, 174, **174**

- OPENOPT keyword, 42*t*
- operand descriptions. *See* OPDESC
- operation code (operations/opcode), 4, 201–426, 441–457
- arguments for, 28, 28*t*
 - CEEDOD (Get Operation Descriptor), **649**
 - free-format syntax and, 566–569, **567**, **569**
 - modern usage of, 441–457. *See also* structured programming
 - optimization and, 443–445
 - structured programming and, 443–457, 469. *See* structured programming
- operators, 106, 106*t*
- Boolean, 220, 220*t*
 - priority of, 107–108, 107*t*
- optimization, 443–445
- OPTIMIZE keyword, 43*t*
- Create Bound Program (CRTBNDRPG) and, 612
- OPTION keyword, 43*t*, 67*t*
- Create Bound Program (CRTBNDRPG) and, 609–611
 - prototyped parameters and, 516–517*t*
- OR, 106, 207*t*, 457
- %BITOR, 134–135, **135**, 135*t*
 - conditioning and branching operations using, 451*t*
 - conditioning indicators not supported by, 209
 - DOW and, 275–277, 275*t*, **276**, **277**
 - fixed-format in, 220, 220*t*
 - IF and, 302–304, 302*t*, **303**, **304**, 472–473, **473**
 - syntax diagram for, 217*t*
 - WHEN and, 402–403, 402*t*, **403**
- OS/400, xxii, xxiii, 429, 486
- integer data type capacity limit in, 57
- OTHER, 207*t*, 341, 341*t*, **341**, 457
- conditioning and branching operations using, 451*t*
 - conditioning indicators not supported by, 209
 - free-format syntax in, 221*t*, 566
 - indicator use and, 445, 447
 - SELECT and, 373–374, 373*t*, **373**
 - syntax diagram for, 217*t*
 - WHEN and, 402–403, 402*t*, **403**
- OTHERWISE, 465
- OUT (Output Data), 207*t*, 263, 342, 342*t*, **342**
- data areas and, 501, 503–505, **504**, **505**
 - free-format syntax in, 221*t*
 - syntax diagram for, 217*t*
- output, 9
- control values in, 28, 28*t*
 - edit codes and, 96–98, 97–98*t*
 - exception (EXCEPT), 9, 88, 206*t*, 216*t*, 250, 351–361, 530*t*, 543*t*, **545**
 - to external data area (OUT), 207*t*, 217*t*, 221*t*, 263, 342, 342*t*, **342**, 501, 503–505, **504**, **505**
 - zero suppression in, 93
- OUTPUT parameter and keywords
- Create Bound Program (CRTBNDRPG) and, 612
- Output specification, 4, 7–8, 7*t*, 37, 37*t*, 81, **82**, 82–90*t*
- externally described files and, 88, 88–89*t*, 90*t*
 - program-described files and, 82, 82–84*t*, 84, 84–88*t*
- OV indicator, 15*t*
- overflow indicators, 18
- OVERLAY keyword, 67*t*
- ## P
- P operation extender, 213
- packed decimal data, 12*t*, 620, 626–627, 627*t*, 628*t*
- convert to (%DEC), , 121*t*, 142–144, **143**, **144**, **573–574**, 574, 588
 - convert to (%DECH), 121*t*, 142–144, **143**, **144**
- PACKEVEN keyword, 68*t*
- %PADDR, 120, 124*t*, 175–175, **175**
- return values from, at compile time, 119
- PAGE reserved word, 26*t*
- PAGEn reserved word, 26*t*
- parameter passing, 486–494
- calling in, 486–487, **486**
 - CEEDOD (Retrieve Operational Descriptor) and, 524, **524**, **525**
 - data structures and, 498
 - data structures passed as, 493–494, **493**, **494**
 - declaring parameter for (PARM), 207*t*, 217*t*, 209, 343–346, 343*t*, **345**, **346**, 491–493, 588, 590–592, 591*t*, **591**, 594–595, 594*t*, **594**, **595**
 - entry parameter list (entry PLIST) and, 487
 - keywords, for prototyped calls/procedures, 500
 - movement of, in calling/called programs, 491–493, **492**, **493**, 493*t*
 - named parameter lists in, 490–493, **490**
 - OPDESC keyword and, 523–524
 - parameter list declaration for (PLIST), 9, 207*t*, 209, 345, 346–348, 346*t*, **347**, 348*t*, 487–493, **493**
 - pointers and, 488, 499
 - prototyped calls in, 489, 494–500, **495**, **496**, **498**. *See also* calling
 - by reference, 488–489, **488**
 - by value, 488
 - weak parameter module in, 489
- parameters
- %BITAND, 131–132, **131**, **132**, 132*t*

- %BITNOT, 133–134, **133**, 134*t*
- %BITOR, 134–135, **135**, 135*t*
- %BITXOR, 136–137, **136**, 137*t*
- Create Bound Program (CRTBNDRPG) and, 605
- declaring (PARM), 207*t*, 217*t*, 209, 343–346, 343*t*, **345**, **346**, 491–493, 588, 590–592, 591*t*, **591**, 594–595, 594*t*, **594**, **595**
- expressions and, 113, 114–115, **114**, **115**
- get count of (%PARMS), , 119–120, 124*t*, 176, **176**, 344, 345
- limits to, 10*t*
- named vs. unnamed, 346–347
- parameter list declaration for (PLIST), 9, 207*t*, 209, 345, 346–348, 346*t*, **347**, 348*t*, 487–493, **493**
- pointers to, 344
- parentheses
 - free-format syntax and, 584
 - operator priority using, 107–108, 107*t*
- PARM (Parameter Declaration), 207*t*, 343–346, 343*t*, **345**, **346**, 491–493
 - conditioning indicators not supported by, 209
 - QtmhPutEnv (put environment value) and, 590–592, 591*t*, **591**
 - QtmhRdSTin (read from standard input/stdin) and, 594–595, 594*t*, **594**, **595**
 - syntax diagram for, 217*t*
 - Web (CGI) programming interfaces and, 588
- %PARMS, 119–120, 124*t*, 176, **176**, 344, 345
- *PARMS, 344
- PASS, 50*t*
- *PDA , 28*t*
- PERRCD keyword, 68*t*, 100
- personal computers (PC) , 552. *See also* workstations
- PGM parameter and keywords
 - Create Bound Program (CRTBNDRPG) and, 606
- PGMNAME, 50*t*
- physical file, 531, 532–537
 - mapped to logic file, 538–542, **538**
- PL/I, xxii, 4, **4**, 37, 108, 443
- *PLACE, 29*t*
- PLIST (Parameter List), 9, 207*t*, 345, 346–348, 346*t*, **347**, 348*t*, 487–493, **493**
 - conditioning indicators not supported by, 209
 - entry parameter list (entry PLIST) and, 487
 - syntax diagram for, 217*t*, 217
- PLIST keyword, 50–51*t*
- pointers, pointer data type, 12*t*, 344, 488
 - activate bound service program (QleActBndPgm), 175
 - %ADDR (get address), 119, 120, 120*t*, 129–130, **130**
 - BASED keyword and, 499
 - Get ILE Exported Item (QleGetExp), 175
 - parameter passing and, 488, 499. *See also* parameter passing
 - resolve system (RSLVSP), 175
- POST, 207*t*, 349–350, 349*t*, **350**, 555, **555**
 - file processing using, 530*t*
 - free-format syntax in, 221*t*
 - syntax diagram for, 217*t*
 - workstation device files and, 552*t*
- power function (**), 107, 108
- powers, syntax of, 108
- precision rules for expressions, 115–116
- PREFIX keyword, 51*t*, 68*t*, 70, 502
- preprocessor directives, 30–33, 30–31*t*, **31–32**
- primary files, limits to, 10*t*
- PRINT key, 21
- printing and printer files, 84
 - limits to, 10*t*
 - print formatted dump (DUMP), 206*t*, 216*t*, 221*t*, 280, 280*t*, **280**
 - skipping output in, 10*t*
 - spacing in, 10*t*
- procedure definition, 91
- Procedure Specification, 4, 7–8, 7*t*, 37, 37*t*, 91–92, **91**, 92*t*
 - keywords in, 92, 92*t*
 - procedure definition in, 91
 - procedures, subprocedures, subroutines and, definition of, 512–514
- procedures, subprocedures, subroutines, 9, 429, 456–457, 465, 507–525. *See also* programs
 - application-specific, 510
 - call bound procedures/programs (CALLB), 205*t*, 215*t*, 241–242, 241*t*, **242**, 343, 344, 345, 347, 429, 433–435, 520, 587, 590–592, 591*t*, **591**. *See also* prototyped calls/procedures
 - prototyped calls/procedures
 - calling, 510–512
 - CALLP, , 105, 113, 119, 205*t*, 220, 220*t*, 221*t*, 243, 243*t*, **244**, 429, 435, 584, 587, 588–592, 589*t*, **589**, **590**, 591*t*, **591**. *See also* prototyped calls/procedures
- CEEDOD (Retrieve Operational Descriptor) and, 524, **524**, **525**
 - conditional or assignment operation calls to, 511
 - Create RPG Module (CRTRPGMOD) and, 510
 - declaring, using free-format syntax, 577–584, **577**, **578**, **579**
 - definition for, 512–514
 - Definition Specification and, 512–514
 - embedded, in multiple source files, 435
 - embedded, in single-source file, 434
 - entry points and, 518–523
 - exported, 509–510, **510**

- procedures, subprocedures, subroutines, *continued*
- expression-based calls to, 511, **511**
 - expressions and, 105
 - EXSR and, 206*t*, 216*t*, 221*t*, 234, 249–295, 294*t*, **295**, **296**
 - external, in multiple source files, 435
 - free-format syntax and, declaring in, 577–584, **577**, **578**, **579**
 - general-purpose, 510
 - get address (%PADDR), 119, 120, 124*t*, 175–175, **175**
 - get address at runtime (GetProcAddr?), 175
 - get parameter count in (%PARMS), , 119–120, 124*t*, 176, **176**, 344, 345
 - hierarchy of, 457, 484–485
 - logical path in, 510
 - mainline, 429, 518
 - modular storage of, 509–510, **509**
 - module definition for, 431–435
 - NOMAIN keyword and, 42*t*, 431, 435, 509, **552–523**
 - operand descriptions (OPDESC) and, 523–524, 648
 - Procedure Specification and, 512–514
 - program entry (PEP) to, 518
 - prototyped, 429, 498, 511–514, **512**, **513**, **514**, 515, 515–517*t*. *See also* prototyped calls/procedures
 - return to calling program from (RETURN), 16, 105, 113, 119, 218*t*, 220, 220*t*, 221*t*, 208*t*, 368–369, 368*t*, **369**
 - return values after calls to, 511
 - single-task only, 457
 - source file source code structure types for, 518–523, 519*t*, **521**, **522–523**
 - source file structure and, 510, **510**
 - storage and, automatic vs. static for, 436–437
 - structured programming and, 465
 - subroutines defined for, 457
 - system interface for, 523–525
 - user entry (UEP), 518
 - user-written, 509–510
- PROCPTR keyword, 68*t*
- product structure file example, 536–537, 537*t*, **537**
- program binding, 429. *See also* binding
- program device
- acquire (ACQ), 205*t*, 221*t*, 215*t*, 228–229, 228*t*, **229**, 332, 364, 530*t*, 552*t*
- program entry procedure (PEP), 518
- program logic cycle, 431
- program status data structure, limits to, 10*t*
- program status data structure (PSDS), 53 , 58, 58*t*, 555, **650**. *See also* POST
- program termination indicator, 16
- *PROGRAM, 314*t*, 337
- program-described files
- file field descriptions in, 73, 73–75*t*, 84, 84–88*t*
 - Input specification in, 69–75, **70**, 71–73*t*, 73–75*t*
 - Output specification and, 82, 82–84*t*
- programming. *See also* modules and modular programming; structured programming
- desk coding and, 6
 - evolution of, xix
 - integrated development environments (IDE) and, 6
 - pseudocode and, 6
 - RPG IV style, 454–457
 - structured. *See* structured programming
 - style of, 454–457, 454
- programming languages, 3, 9, 443
- programs, 205–208*t*, 427–439. *See also* procedures, subprocedures, subroutines; source files
- abbreviations and symbols (tokens) for, 204*t*
 - application source file in, 435
 - binding in, 429, 433–434
 - Boolean, 220, 220*t*
 - brackets in, 214
 - branching within (GOTO), 301, 301*t*, **301**
 - Calculation specification and, 203
 - calling, 433–434
 - bound procedures/programs (CALLB), 205*t*, 215*t*, 241–242, 241*t*, **242**, 343, 344, 345, 347, 429, 433–435, 520, 587, 590–592, 591*t*, **591**. *See also* prototyped calls/procedures
 - dynamic (CALL), 205*t*, 215*t*, 239–240, 239*t*, **239**, **240**, 343, 344, 345, 347, 433–434, 435, 444, 594–595, 594*t*, **595**, **595**
 - with prototype (CALLP), , 105, 113, 119, 205*t*, 220, 220*t*, 221*t*, 243, 243*t*, **244**, 429, 435, 584, 587, 588–592, 589*t*, **589**, **590**, 591*t*, **591**. *See also* prototyped calls/procedures
 - case sensitivity in, 214
 - column numbers/placement of elements in, 203
 - compare and branch (CAB), 205*t*, 215*t*, 220, 220*t*, 238, 238*t*, **238**, 284–285, 284*t*, **285**, 389, 389*t*, **389**, 451*t*, 467*t*, 478–479, 479*t*
 - conditioning indicator and, 203, 209, 209*t*
 - control-level indicator and, 203
 - Create Bound Program (CRTBNDRPG) and, 430, 434, 605–618
 - Create Program (CRTPGM) and, 434
 - Create RPG Module (CRTRPGMOD) and, 430, 434, 605
 - date, time, timestamp format codes in, 209–210, 210*t*, 211*t*
 - decimal positions in, 203
 - design of, in structured programming, 462

- directories and subdirectories for, 430–431
 - duration codes for, 211–212, 211*t*
 - entry points in, 465
 - exception (EXCEPT), 9, 88, 206*t*, 216*t*, 250, 351–361, 530*t*, 543*t*, **545**
 - exit points in, 465, 484
 - EXPORT keyword and, 437–439, **438**, **439**
 - extended factor 2 in, 203
 - extender reference for, 212, 213*t*
 - factor 1 and factor 2 in, 203
 - field length in, 203
 - free-format code in, 220–221, 220*t*, 221*t*
 - hierarchy chart for, 485
 - IMPORT keyword and, 437–439, **438**, **439**
 - Integrated File System (IFS) and, 429–431
 - label defined in (TAG), 9, 11, 208*t*, 209, 218*t*, 389, 389*t*, **389**, 447, 478, 479, 479*t*
 - looping in . *See* looping
 - machine readable object or executables in, 433
 - mainline procedures in, 429, 431, **431**
 - modules in, 431–435. *See also* modular programming
 - multiple source files with embedded or external subprocedures, 435
 - natural expressions in, 220, 220*t*, 222
 - nested code in, 227, **227**
 - organization of, 427–439
 - program logic cycle and, 431
 - program status error codes in, 184*t*, 222, **222**, 223*t*, 337*t* prototyped, 512
 - qualified names and, 240
 - read in a data area (IN), , 206*t*, 216*t*, 221*t*, 263, 305, 305*t*, **305**, 501, 503–505, **504**, **505**
 - redundant code in, 465, 484
 - result field in, 203
 - resulting indicators in, 203
 - return to calling (RETURN), 16, 105, 113, 119, 218*t*, 220, 220*t*, 221*t*, 208*t*, 368–369, 368*t*, **369**
 - scope and, 435–436, **436**
 - single-source file (traditional) in, 434
 - single-source file with embedded subprocedures in, 434
 - source files in, 429–431
 - special device files and, 348, 348*t*
 - status code for (%STATUS), 125*t*, 155, 182–184, **182**, 183*t*, 222, 224, 413–415*t*, 420–422*t*
 - status error codes in, , 184*t*, 222–226, **222**, 223*t*, 337*t*
 - storage and, automatic vs. static for, 436–437
 - subprocedures and, 429
 - syntax of, 214, 215–219*t*
 - top-down development in, 462–465, **463**, **464**
 - unconditional , 209, 209*t*
 - values in, 214
 - variables in, 214, 431
 - prototyped calls/procedures, 429, 435, 489, 494–500, **495**, **496**, **498**
 - calling (CALLP), , 105, 113, 119, 205*t*, 220, 220*t*, 221*t*, 243, 243*t*, **244**, 429, 435, 584, 587, 588–592, 589*t*, **589**, **590**, 591*t*, **591**. *See also* prototyped calls/procedures
 - data structures and, 498–500, **500**
 - keywords for, 500, 515–517*t*
 - programs and, 512
 - PRTCTL keyword, 52*t*
 - pseudocode, 6
 - *PSSR *t*, 28*t*, 222
- ## Q
- QC2LE binding directory, 601
 - QHTTPSVR library, Web (CGI) programming interfaces and, 588
 - QILE activation group, 434
 - QleActBndPgm, 175
 - QleGetExp, 175
 - QtmhCvtDB (convert to database structure) in, 587*t*, 596–600, 598–599*t*, **599**, **600**
 - QtmhGetEnv (get value of environment variable) , 587*t*, 588–590, 589*t*, **589**, **590**
 - QtmhPutEnv (put environment value) in, 587*t*, 590–592, 591*t*, **591**
 - QtmhRdSTin (read from standard input/stdin) in, 587*t*, 594–595, 594*t*, **594**, **595**
 - QtmhWrStout (write to standard input/stdout) in, 587*t*, 595–596, 596*t*, **596**
 - QUALIFIED keyword, 11, 68*t*
 - free-format syntax and, 576, 576*t*
 - qualified names, 240
 - naming conventions for, 11*t*
 - QUERY_STRING , 592, 593*t*
 - QZHBCGI service program, 588
 - QzhhCgiParse, 587*t*
 - QzhhCgiRecvState, 587*t*
 - QzhhCgiSendState, 587*t*
 - QzhhCgiUtils, 587*t*
- ## R
- R operation extender, 213*t*
 - r-value , 108–110, **109**, 288
 - RAFDATA keyword, 52*t*

- random access file processing, 531, 533, 553, **553**
 CHAIN operation and, 553, **553**
- ranges, 453–454, **453, 454**
- READ (Read Record), 19, 207*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352, 353**, 529, 543, 556, **556**
 cursor position in file after execution of, 543*t*, **547**
 decimal data errors and, 623
 EXFMT and, 206*t*, 216*t*, 221*t*, 293, 293*t*, **293**, 530*t*, 552*t*, 554–555, **554**
 file processing using, 530*t*
 free-format syntax in, 221*t*
 key list in (%KDS), 123*t*, 128, 168–169, **169**
 QtmhRdSTin (read from standard input/stdin) and, 594–595, 594*t*, **594, 595**
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358, 359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- SETLL and, 123*t*, 128, 168–169, **169**, 208*t*, 218*t*, 221*t*, 250, 376–377, 376*t*, **377**, 530*t*, 543*t*, **548**
 syntax diagram for, 217*t*
 timeout and, 353, 556
- READC (Read Next Changed Record), 207*t*, 354, 354*t*, **354, 557, 557**
 file processing using, 530*t*
 free-format syntax in, 221*t*
 syntax diagram for, 217*t*
 workstation device files and, 552*t*
- READE (Read Next with Equal Key), 207*t*, 250, 355–356, 355*t*, **356**, 543, 550
 cursor position in file after execution of, 543*t*, **547**
 file processing using, 530*t*
 free-format syntax in, 221*t*
 keylists and KLIST in, 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308, 530*t***
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352, 353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594, 594, 595**, 623
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358, 359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
 record-lock contention caused by, 544
 syntax diagram for, 217*t*
- READP (Read Prior), 207*t*, 250, 268, 357–359, 357*t*, **358, 359**, 543
 cursor position in file after execution of, 543*t*, **547**
 file processing using, 530*t*
 free-format syntax in, 221*t*
 keylists and KLIST in, 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308, 530*t***
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352, 353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594, 594, 595**, 623
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- SETLL and, 123*t*, 128, 168–169, **169**, 208*t*, 218*t*, 221*t*, 250, 376–377, 376*t*, **377**, 530*t*, 543*t*, **548**
 syntax diagram for, 217*t*, 218*t*
- READPE (Read Prior with Equal Key), 207*t*, 250, 360–361, 360*t*, **361**, 543
 cursor position in file after execution of, 543*t*, **547**
 file processing using, 530*t*
 free-format syntax in, 221*t*
 key list in (%KDS), 123*t*, 128, 168–169, **169**
- READ and, 351–353, 351*t*, **352, 353**
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358, 359**, 530*t*, 543, 543*t*, **547**
 record-lock contention caused by, 544
 syntax diagram for, 218*t*
- REALLOC (Reallocate Memory), 207*t*, 362–363, 362*t*, **363**
 syntax diagram for, 218*t*
- %REALLOC, 125*t*, 262, 362
- RECNO keyword, 52*t*
- record address files, limits to, 10*t*
- record identifying entries, 69. *See also* Input specifications
- record-identifying indicators, 18
- records, 531. *See also* databases; files
 delete (DELETE), 88, 157–158, 157*t*, **158**, 168–169, **169**, 205*t*, 215*t*, 221*t*, 250, 266–268, 266*t*, **267, 268**, 308, 308*t*, 355–361, 530*t*, 543*t*, **545**
 force end of data (FEOD), 206*t*, 216*t*, 221*t*, 250, 297, 297*t*, **297**, 351–361, 530*t*, 543*t*, **546**
 force input priority (FORCE), 206*t*, 216*t*, 221*t*, 300, 300*t*, **300**
 limits to, 10*t*, 10
 random access/read of (CHAIN), 19, 157–158, 157*t*, **158**, 168–169, **169**, 205*t*, 215*t*, 221*t*, 224, 249–251, 249*t*,

- 250–251**, 268, 308, 308*t*, **308**, 351–361, 530*t*, 533, **533**, 553, **553**, 543*t*, **545**, 553*t*, 563, **563**, 623
 read from (READ), 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
 read next changed (READC), 207*t*, 217*t*, 221*t*, 354, 354*t*, **354**, 530*t*, 552*t*, 557, **557**
 read next with equal key (READE), 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
 read prior from data file (READP), 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
 read prior with equal key (READPE), 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
 READE and READPE and, record-lock contention caused by, 544
 set greater than (SETGT) , 9, 123*t*, 128, 168–169, **169**, 157–158, 157*t*, **158**, 206, 208*t*, 209, 218*t*, 221*t*, 250, 374–375, 374*t*, **357**, 530*t*, 543*t*, **548**
 set lower limit (SETLL), 123*t*, 128, 168–169, **169**, 208*t*, 218*t*, 221*t*, 250, 376–377, 376*t*, **377**, 530*t*, 543*t*, **548**
 unlock (UNLOCK), 208*t*, 218*t*, 221*t*, 250, 263, 292, 398–399, 398*t*, **399**, 501, 503–505, **504**, **505**, 530*t*, 543*t*
 update (UPDATE), 88, 156, **156**, 208*t*, 219*t*, 221*t*, 250, 400–402, 400*t*, **401**, **402**, 530*t*, 543*t*, **548**, 552*t*557–558, **558**
 write to (WRITE), 88, 208*t*, 219*t*, 221*t*, 404–405, 404*t*, **405**, 529, 530*t*, 543*t*, **548**, 552*t*, 554–555, **554**, 558, **559**
- redundant code, 465, 484
 REL (Release Record), 207*t*, 292
 file processing using, 530*t*
 free-format syntax in, 221*t*
 syntax diagram for, 218*t*
 workstation device files and, 552*t*
 relational database management systems, 623
 relationship testing, using IF, 472–474, **472**, **473**
 %REM, 125*t*, 177, **177**
 remainders
 move remainder of division (MVR), 207*t*, 217*t*, 331, 331*t*, **331**
 %REM, 125*t*, 177, **177**
 RENAME keyword, 52*t*
 REPLACE parameter and keywords
 Create Bound Program (CRTBNDRPG) and, 614–615
 %REPLACE, 125*t*, 178
 Report Generator Language. *See* RPG
 REQUEST_METHOD , 592, 593*t*, 595
- reserved names, 26, 26*t*
 reserved words, 27, 27*t*
 RESET, 208*t*, 257
 free-format syntax in, 221*t*
 syntax diagram for, 218*t*
 result fields, 3, 203
 clear (CLEAR), 205*t*, 215*t*, 221*t*, 256–258, 256*t*, 257*t*, **258**, 317, 366, 425, 626
 copy data (MOVE), 3, 4, 20, 207*t*, 209, 210, 212, 217*t*, 222, 316–319, 316*t*, 318*t*, **318**, **319**, 378, 379, 444, 568, **569**, 570–571, 570*t*, **571**, 571*t*, 580*t*
 copy data (MOVE), 207*t*, 209, 212, 271*t*, 281, 324–328, 324*t*, **325**, **326**, 327*t*, **328**, 344, 568, **569**, 570–571, 570*t*, **571**, 571*t*
 reset variables to initial value (RESET), 365–367, 365*t*, **367**
 retrieve system time/date (TIME), 396–397, 396*t*, 397*t*, **397**
 test (TEST), 16, 208*t*, 209, 218*t*, 221*t*, 390–391, 391*t*, **391**, 580*t*
 test zone of (TESTZ), 208*t*, 218*t*, 395, 395*t*, **395**
 zero and add (Z-ADD), 26, 208*t*, 219*t*, 344, 425, 425*t*, **425**
 zero and subtract (Z-SUB), 208*t*, 219*t*, 426, 426*t*, **426**
- resulting indicators, 19
 programs, 203
 RETURN (Return a Value), 16, 105, 113, 208*t*, 368–369, 368*t*, **369**
 built-in functions and, 119
 free-format syntax in, 221*t*
 natural expression support in, 220, 220*t*
 syntax diagram for, 218*t*
 Return-as-Uppercase source code, **645**
 return codes, special device files and, 348, 348*t*
 Return Length of Input String source code, **647**
 return points, subroutine, 29–30, 29–30*t*
 return values
 %ABS (absolute value), 129, **129**
 built-in functions and, 119
 expressions and, 113
 procedures, subprocedures, subroutines and, after calls to, 511
 %SIZE, 119, 120, 125*t*, 180–181, 180*t*, **181**
 reusable code, 484, 485
 right-signed data type, 12*t*
 ROLBK (Rollback Transaction), 208*t*, 370, 370*t*, **370**
 file processing using, 530*t*
 free-format syntax in, 221*t*
 syntax diagram for, 218*t*
 ROLLDOWN key, 21
 ROLLUP key, 21
 roots, syntax of, 108

- routines. *See also* procedures, subprocedures, subroutines
 reserved words in, 27, 27*t*
 structured programming and, well-formed, 465–466
- RPG, xix, xx, 1–33
 arguments for operation codes in, 28, 28*t*
 arrays in, 9
 brevity of, 6–7, 7
 built-in functions in, 117–200. *See also* built-in functions
 Calculation specification in, 4, 7–8, 7*t*, 12, 37, 37*t*, 78, 78–81*t*, 112–113, **113**, 203, 565–570
 calculation statements in, 4, 9
 case statements (in-line) in, 5
 columnar expression format (hybrid free format) code in, 3
 compilers and, 30–33, 30–31*t*, **31–32**
 components of, 9
 constants in, 21–27
 Control specification in, 7–8, 7*t*, 37, 37*t*
 cycle of. *See* RPG cycle, 300
 data structures in, 9
 data types in, 12, 12*t*
 Definition specification in, 4, 7–8, 7*t*, 12, 37, 37*t*, 53, 53–69*t*, 112, **112**
 directives in, 30–33, 30–31*t*, **31–32**
 conditional compiler preprocessor source, 32–33, 33*t*
 editing numeric output in, 92–99
 editors for, 4, 6
 enhanced or alternate calculation specifications syntax, 563
 evolution of, xx–xxii
 expressions in, 103–116
 extended name specification in, **53**, 55–56, 56*t*, **56**
 field definition specification in, **54**
 fields in, 9
 figurative constants in, 21–27, 23–24*t*
 File specification in, 4, 7–8, 7*t*, , 37, 37*t*, 44–53, 44–53*t*
 files in, 9
 fixed-format code in, 3, 563, 564, **564**
 free-format code in, 3, 4, **4**, 6, 561–584
 function key attention identification bytes in, 17–18, 17*t*, **18**, **650**
 Header specification in, 4, 8*t*, 37, 37*t*, 38–44, 38–44*t*
 indicatorless code in, 14, **14**, 15–16, 447–448, **447**
 indicators in, 13–15, **13**, 14–15*t*, 15–16, 18–20, 443–446, 455
 input in, 9
 Input specification in, 4, 7–8, 7*t*, 37, 37*t*, 69–77
 labels in, 9
 limitations of, 9, 10*t*
 literal values in, 21–27, 23*t*
 logic and indicator use in, 445–446, **445**, **446**
 named constants in, 9, 21–27, **25**
 naming conventions in, 11, 11*t*
 output control values in, 28, 28*t*
 output in, 9
 Output specification in, 4, 7–8, 7*t*, 37, 37*t*, 81, **82**, 82–90*t*
 position-oriented program specifications in, 6. *See also* specifications in RPG
 Procedure specification in, 4, 7–8, 7*t*, 37, 37*t*, 91–92, **91**, 92*t*
 procedures in, 9
 program status data structure (PSDS) and, 53, 58, 58*t*, 555, **650**. *See also* POST
 reserved names in, 26, 26*t*
 reserved words in, 27, 27*t*
 special reserved position (star position) specification in, **54**, 53
 specification sequence in, 8, 8*t*, 37, 37*t*
 specifications in, 4, 6, 7–8, 7*t*. *See also* specifications in RPG
 formats for, 35–101. *See also* specifications in RPG
 structured programming and, 443–457, 461
 subroutines in, and return points for, 29–30, 29–30*t*
 versions of, covered in book, xxii
- RPG cycle, xxi–xxiv, 19, 88, 435
 decimal data errors and, 623
 entry points in, 487, 518–523
 force input priority (FORCE), 206*t*, 216*t*, 221*t*, 300, 300*t*, **300**
 force next input (NEXT), 207*t*, 217*t*, 221*t*, 332–333, 332*t*, **333**
- RPG IV, xxii–xxiv, 3, 454–457, 461
 RPG xTools date routines, 639–641, 639–641*t*
 RPG/400 compiler, xxi. *See also* compilers
 RSLVSP, 175
 RT indicator, 15*t*, 16
 Russell, Bertrand, 105
- S**
- S/3, xix
 S/370, xix
 SAA RPG Level 1, xxi
 SAVEDS keyword, 52*t*, 332
 SAVEIND keyword, 52*t*
 SCAN (Scan String for Value), 16, 208*t*, 371–372, 371*t*, **372**
 %FOUND versus indicators in, 157–158, 157*t*, **158**
 syntax diagram for, 218*t*, 218
 %SCAN, 125*t*, 179, **179**
 scope, 435–436, **436**
 search
 arrays (element lookup) %LOOKUP, 124*t*, 171, **171**
 lookup array or table (LOOKUP), 16, 122*t*, 207*t*, 217*t*, 154–155, 154*t*, **155**, 311–312, 311*t*, **312**, 376, 377, 444

- SCAN, 16, 157–158, 157*t*, **158**, 208*t*, 218*t*, 218, 371–372, 371*t*, **372**
- *SECONDS , 146*t*, 187, 211*t*, 231, 296
- %SECONDS, 125*t*, 148
- SELECT, 208*t*, 373–374, 373*t*, **373**, 457, 465, 467*t*, 539–540
- END and, 283, 283*t*, **283**
- free-format syntax in, 221*t*, 566
- indicator use and, 444, 445, 447
- nested, 10*t*, 227
- OTHER and, 341, 341*t*, **341**
- syntax diagram for, 218*t*
- WHEN and, 402–403, 402*t*, **403**
- select/omit, testing for, 453–454, **453**, **454**
- SELECT-WHEN, 5, 5, 21
- CASE structures and, 475, **475**
- SELECT-WHEN-OTHER, 467
- compare and perform subroutine (CAS), 21, 205*t*, 215*t*, 220, 220*t*, 234, 245–247, 245*t*, **245**, **246**, 283, 283*t*, **283**, 444, 451*t*, 467*t*
- selection (choice), in structured programming, 466, 467, 467*t*
- semicolon statement-terminator, free-format syntax and, 565, 567, 584
- separator (edit character) for dates/times, 634–635
- sequence, in structured programming, 466, **466**
- sequential files, 543
- SETGT (Set Greater Than), 208*t*, 250, 374–375, 374*t*, **357**
- cursor position in file after execution of, 543*t*, **548**
- file processing using, 530*t*
- %FOUND versus indicators in, 157–158, 157*t*, **158**
- free-format syntax in, 221*t*
- key list in (%KDS), 123*t*, 128, 168–169, **169**
- keylists and KLIST in, 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308**, 530*t*
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- syntax diagram for, 218*t*
- SETLL (Set Lower Limit), 208*t*, 250, 376–377, 376*t*, **377**
- cursor position in file after execution of, 543*t*, **548**
- %EQUAL and, 122*t*, 154–155, 154*t*, **155**, 311, 376, 377
- file processing using, 530*t*
- %FOUND versus indicators in, 157–158, 157*t*, **158**
- free-format syntax in, 221*t*
- key list in (%KDS), 123*t*, 128, 168–169, **169**
- keylists and KLIST in, 9, 206*t*, 209, 217*t*, 307, 308, 308*t*, **308**, 530*t*
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- syntax diagram for, 218*t*
- SETOFF (Set Bit Off), 208*t*, 378, 378*t*, **378**
- syntax diagram for, 218*t*
- SETON (Set Bit On), 20, 208*t*, 379, 379*t*, **379**
- syntax diagram for, 218*t*
- SEU Source Editor, 6
- SFILE keyword, 52*t*
- SHTDN (Shut Down System), 208*t*, 380, 380*t*, **380**
- syntax diagram for, 218*t*
- %SHTDN, 125*t*, 179
- sign or status, decimal data errors and, 624
- signed data type, 12*t*
- signed integers, 56–58, 57*t*, 58*t*
- %SIZE, 119, 120, 125*t*, 180–181, 180*t*, **181**
- return values from, at compile time, 119
- SLN keyword, 53*t*
- SORT utility, 21
- SORTA (Sort Array), 186, 208*t*, 381–382, 381*t*, **381**, **382**
- free-format syntax in, 221*t*
- natural expression support in, 220, 220*t*
- syntax diagram for, 218*t*
- sorts
- ASCEND and DESCEND keywords in, 381–382
- SORT utility for, 21
- SORTA, 186, 208*t*, 218*t*, 220, 220*t*, 221*t*, 381–382, 381*t*, **381**, **382**
- source code listings, 643–650
- source fields, 3
- source files. *See also* procedures, subprocedures, subroutines; programs
- application source file in, 435
- compiler action on, 433
- EXPORT keyword and, 437–439, **438**, **439**

- source files., *continued*
- flat-file systems and, 429–431
 - IMPORT keyword and, 437–439, **438**, **439**
 - Integrated File System (IFS) and, 429–431
 - location of, 429–431
 - mainline procedures in, 431, **431**
 - member contents of, 431
 - multiple source, with embedded/external subprocedures
 - procedures, subprocedures, subroutines and, 510, **510**
 - source code structure types for, 518–523, 519*t*, **521**, **522–523**
 - single-source, 434
 - single-source with embedded subprocedures, 434
 - structure of, 431–435, **431**, 434*t*
- /SPACE directive, 31*t*
- spaghetti code, 447
- SPECIAL device files, 347–348, 348*t*
- special reserved position (star position) specification, **54**
- specifications in RPG, 4, 6, 7–8, 7*t*, 35–101
- arrays and, Definition specification for, 100–101, **101**
 - Calculation specification in, 7–8, 7*t*, 12, 37, 37*t*, 78, 78–81*t*, 112–113, **113**, 203, 565–570
 - Control specification, 7–8, 7*t*, 37, 37*t*
 - Definition specification, 7–8, 7*t*, 12, 37, 37*t*, 112, **112**
 - descriptions of, 8*t*, 37, 37*t*
 - edit words in, 92–94, 93*t*, **94**
 - editing numeric output and, 92–99
 - extended name, **53**, 55–56, 56*t*, **56**
 - field definition specification in, **54**, 53
 - File specification, 7–8, 7*t*, 37, 37*t*, 44–53, 44–53*t*
 - formats for, 35–101
 - Header specification, 8*t*, 37, 37*t*, 38–44, 38–44*t*
 - Input specification, 7–8, 7*t*, 37, 37*t*, 69–77
 - input/output feedback data structure (INFDS) in, 59, 59–60*t*
 - integer data types in, 56–58, 57*t*, 58*t*
 - Output specification, 7–8, 7*t*, 37, 37*t*, 81, **82**, 82–90*t*
 - position-oriented, 6
 - Procedure specification, 7–8, 7*t*, 37, 37*t*, 91–92, **91**, 92*t*
 - program status data structure (PSDS) and, 53, 58, 58*t*, 555, **650**. *See also* POST
 - sequence of, 8, 8*t*, 37, 37*t*
 - special reserved position (star position) specification in, **54**
 - tables and, Definition specification for, 100–101, **101**
 - types of, 37, 37*t*
- SQL, 21
- directives in, 30–33, 30–31*t*, **31–32**
 - File specification and, 44
 - free-format syntax and, 584
- SQRT (Square Root), 208*t*, 222, 382–383, 382*t*, **383**
- free-format syntax and, 569–570, **569**, **570**
 - indicator use and, 444
 - syntax diagram for, 218*t*
- %SQRT, 125*t*, 182
- SRCFILE parameter and keywords
- Create Bound Program (CRTBNDRPG) and, 606
- SRCMBR parameter and keywords
- Create Bound Program (CRTBNDRPG) and, 606
- SRTSEQ parameter and keywords
- Create Bound Program (CRTBNDRPG) and, 614
- *SRVPGM service program, 588
- standard input (stdin)
- read from (QtmhRdStIn), 594–595, 594*t*, **594**, **595**
- standard output (stdout)
- QtmhWrStout (write to standard input/stdout) and, 595–596, 596*t*, **596**
- start values, for loops, 298
- *START, 28*t*
- STATIC keyword, 69*t*, 437
- static storage (memory), 436–437
- status codes, 222, **222**, 223*t*
- file or program (%STATUS), 125*t*, 155, 182–184, **182**, 183*t*, 222, 224, 413–415*t*, 420–422*t*
 - file error codes in, 183*t*, 224–225, **224**, 225–226*t*, 337*t*
 - program error codes in, 184*t*, 222, **222**, 223*t*, 337*t*
- Status Data Structure (SDS), get parameter count in (%PARMS), 119–120, 124*t*, 176, **176**, 344, 345
- %STATUS, 125*t*, 155, 182–184, **182**, 183*t*, 222, 224
- XML extended exception/error codes for, 413–415*t*, 420–422*t*
- status, 95*t*
- stdin. *See* standard input
- storage. *See* memory
- %STR, 125*t*, 128, 185, **185**
- string and substring data
- CEEGSI (Get String Information), 648, **648**
 - concatenate (CAT), 205*t*, 215*t*, 247–248, 247*t*, **248**
 - Convert to Uppercase source code for, **646**
 - convert to UCS2 (%UCS2), 193
 - date (%SUBDT), 126*t*, 187, 211, 296, 580
 - evaluate, right-adjust (EVALR), 105, 108–110, **109**, 116, 119, 206*t*, 220, 220*t*, 221*t*, 288–289, 288*t*, **289**, 570–571, 570*t*, **571**, 571*t*
 - evaluation (EVAL), 4, 105, 108–110, **109**, 119–120, 120*t*, 129–130, **130**, 206*t*, 216*t*, 220, 220*t*, 221*t*, 222, 286–287, 286*t*, **287**, 287*t*, 378, 379, 382, 388, 569–571, **569**, **570**, 570*t*, **571**, 571*t*, 584
 - extract portion of (%SUBDT), 126*t*, 187, 211, 296, 580

- get/set substring of a value in (%SUBST), 109, 120, 126*t*, 128, 187–189, **188–189**, 288, 387
- get/store null-terminated C-language based (%STR), 185, **185**
- GetDateAsString RPG routine for, 641*t*
- move with array (MOVEA), 207*t*, 217*t*, 222, 322–323, **323**, 323*t*, 378, 379
- replace/insert (%REPLACE), 178
- Return as Uppercase source code for, **645**
- Return Length of Input String source code for, **647**
- scan (%SCAN), , 125*t*, 179, **179**
- scan (SCAN), 16, 157–158, 157*t*, **158**, 208*t*, 218*t*, 218, 371–372, 371*t*, **372**
- substring then move left (SUBST), 218*t*, 387–388, 387*t*, **388**
- translate (XLATE), 208*t*, 219*t*, 407–409, 407*t*, **408**, **409**
- trim (%TRIM, %TRIML, %TRIMR), 120, 126*t*, 192–193, **192–193**
- verify (CHECK), 157–158, 157*t*, **158**, 205*t*, 215*t*, 251–253, 251*t*, **252**, **253**
- verify right to left (CHECKR), 157–158, 157*t*, **158**, 205*t*, 215*t*, 254–255, 254*t*, **255**
- ZTools source code listings for, 643–650
- string expressions, 108
- structured analysis, structured programming and, 462
- structured design, structured programming and, 462
- structured programming, xxi, 443–457, 459–479
 - binding in, 466
 - Boolean operations in, 470, 470*t*
 - branching in, 451–452, 451*t*, **452**
 - calling in, 484
 - CASE structure and, 474–479, **474**
 - compare (COMP) and, 16, 119, 205*t*, 215*t*, 261, 261*t*, 444, **261**, 444, 451*t*, 452–454, **452**, **453**, 457, 467, 467*t*
 - compare and branch operations in, 451–452, 451*t*, **452**
 - control flow constructs in, 466–469
 - converting legacy code to, 457
 - development and history of, 461–470
 - DO loops in, 448–450, **448**, **449**, **450**
 - entry points in, 465, 487, 518–523
 - exit points in, 465, 484
 - flag setting in, 465
 - hierarchy chart in, 484–485, **485**
 - IF-THEN-ELSE in, 470–474
 - indicator use in, 13, 15–16, 443–446, 455
 - indicatorless code and, 447–448, **447**
 - iteration in, 466, 467–469, **468**, 468*t*, **469**
 - learning, 461–462
 - level-break processing and, 449–450, **449**
 - lists in, testing for, 453–454, **453**, **454**
 - logic and indicator use in, 445–446, **445**, **446**
 - logic control in, 465, 474, 484
 - looping in, 465, 466, 467–469, **468**, 468*t*, **469**, 484. *See also* looping
 - math operation codes and, 456, **456**
 - modular programming and, 481–505, 510. *See also* modules and modular programming
 - naming conventions and, 463
 - object-based programming as basis for, 461
 - operation code in, 469
 - optimization and, 443–445
 - outlining the processes for, 463–464, **464**
 - PLIST and parameter passing in, 9, 207*t*, 209, 345, 346–348, 346*t*, **347**, 348*t*, 487–493, **493**
 - program design and, 462
 - prototyped calls and, 489, 494–500, **495**, **496**, **498**
 - ranges in, testing for, 453–454, **453**, **454**
 - redundant code and, 465, 484
 - relationship testing in, 472–474, **472**, **473**
 - repetition in, 466, 467–469, **468**, 468*t*, **469**
 - reusable code in, 484, 485
 - routines in, well-formed, 465–466
 - RPG IV style and, 454–457
 - select/omit in, testing for, 453–454, **453**, **454**
 - selection (choice) in, 466, 467, 467*t*
 - sequence in, 466, **466**
 - spaghetti code vs., 447
 - structured analysis in, 462
 - structured design in, 462
 - style and, 454–457, 454
 - subprocedures in, 456–457, 465
 - subroutines in, 456–457
 - top-down design and development in, 455, 462–465, **463**, **464**, 476, 484
- structured query language. *See* SQL
- structures, naming conventions for, 11*t*
- SUB (Subtract), 208*t*, 222, 383–384, 383*t*, **384**, 456
 - free-format syntax and, 569–570, **569**, **570**
 - syntax diagram for, 218*t*
- %SUBARR, 125*t*, 162, 171, 186, **186**
- %SUBBDT, 126*t*, 187, 211, 296, 580
- SUBDUR (Subtract Duration), 208*t*, 211, 212, 385–386, 385*t*, **386**
 - data/time manipulation using, free- vs. fixed-format syntax, 580*t*
 - syntax diagram for, 218*t*
- subprocedures. *See* procedures, subprocedures, subroutines
- Subroutine (EXSR) operation. *See* EXSR

- subroutines. *See also* procedures, subprocedures, subroutines, 456–457
- begin (BEGSR), 9, 205*t*, 209, 215*t*, 221*t*, 234, 234*t*, **234**, 341, 373
- CASE structures and, 474–476, **474**
- compare and perform (CAS), 21, 205*t*, 215*t*, 220, 220*t*, 234, 245–247, 245*t*, **245**, **246**, 283, 283*t*, **283**, 444, 451*t*, 467*t*
- end (ENDSR), 9, 11, 206*t*, 209, 216*t*, 221*t*, 284–285, 284*t*, **285**, 234, 301, 310, 341, 373, 478, 479, 479*t*
- exit (LEAVESR), 217*t*, 221*t*, 207*t*, 294, 310, 310*t*, **310**, 479, 479*t*
- EXSR operation and, 206*t*, 216*t*, 221*t*, 234, 249–295, 294*t*, **295**, **296**
- flow of control through, 295, **295**
- GOTO and, 301, 301*t*, **301**
- level-break indicators and, 295
- limits to, 10*t*
- mainline calculations in, 295, **295**
- perform (EXSR), 206*t*, 216*t*, 221*t*, 234, 249–295, 294*t*, **295**, **296**
- return points for, 29–30, 29–30*t*, 29
- single-task only, 457
- subprocedures vs., 457
- SUBST (Substring Data), 387–388, 387*t*, **388**
- syntax diagram for, 218*t*
- %SUBST, 120, 126*t*, 128, 187–189, **188–189**, 288, 387
- assignment and, 109
- substring data. *See* string and substring data
- subtraction, 106
- short- vs. long-form, 383–384
- SUB, 208*t*, 218*t*, 222, 383–384, 383*t*, **384**, 456, 569–570, **569**, **570**
- zero and subtract (Z-SUB), 208*t*, 219*t*, 426, 426*t*, **426**
- Super Julian dates, 633
- switches, 13, 14–15*t*. *See also* indicators
- symbols in expressions, 108, 204*t*
- syntax. *See also* fixed-format syntax; free-format syntax
- alternate options in, 561–584. *See* free-format code
- built-in functions and, 119–120, 119
- enhanced or alternate calculation specifications, 563 , 564–564, **564**
- fixed-format, 563, 564, **564**
- free-format, 561–584
- programs, 214, 215–219*t*
- *SYS *t*, 210*t*, 211*t*
- system functions/interface, 523–525
- alternative naming for, 524, **524** , **525**
- calling programs and, 498
- procedures, subprocedures, subroutines and, 523–525
- System i5, xxii
- system shut down
- check for (%SHTDN), 179
- test for request (SHTDN), 208*t*, 218*t*, 380, 380*t*, **380**
- System/3, xxi
- System/32, xix, xx
- System/38, xxi
- ## T
- T operation extender, 213*t*
- tables
- compile-time data and, 100–101, **101**
- Definition specification for , 100–101, **101**
- limits of, 10*t*
- lookup (LOOKUP), 16, 122*t*, 207*t*, 217*t*, 154–155, 154*t*, **155**, 311–312, 311*t*, **312**, 376, 377, 444
- number of elements in (%ELEM), 152, **152**
- search for element in (%TLOOKUP), 126*t*, 191
- TAG (Create Tag), 9, 11, 208*t*, 389, 389*t*, **389**, 478
- branching using, 479, 479*t*
- conditioning indicators not supported by, 209
- indicator use and, 447
- syntax diagram for, 218*t*
- target fields, 3
- TEST (Test Result Field), 16, 208*t*, 209, 390–391, 391*t*, **391**
- data/time manipulation using, free- vs. fixed-format syntax, 580*t*
- free-format syntax in, 221*t*
- syntax diagram for, 218*t*
- TESTB, 208*t*, 392–393, 392*t*, **393**
- syntax diagram for, 218*t*
- TESTN, 208*t*, 394, 394*t*, **394**
- syntax diagram for, 218*t*
- TESTZ, 208*t*, 395, 395*t*, **395**
- syntax diagram for, 218*t*
- TEXT keyword, 43*t*
- Create Bound Program (CRTBNDRPG) and, 607
- TGTRLS parameter and keywords
- Create Bound Program (CRTBNDRPG) and, 617
- %THIS, 126*t*
- THREAD keyword, 43*t*
- time, 12*t*. *See also* date; timestamp
- add duration to (ADDDUR), 231, 231*t*, **230**
- %CHAR, 121*t*, 137–138, **138**, 571–572, **571**, **572**, 582
- convert to (%TIME), 126*t*, 189, 189*t*, 572–573, **573**, 581, 582–583, **582**, **583**

- convert to packed decimal (%DEC), , 121*t*, 142–144, **143**, **144**, **573–574**, 574, 588
- convert to timestamp (%TIMESTAMP), 126*t*, 190, 190*t*, 581, 582–583, **582**, **583**
- DATFMT keyword for, 634–635, 634*t*
- DDS layout for, 631–641
- default values for, 633
- difference between (%DIFF), 122*t*, 146, 146*t*, **146**, 211, 580–582
- DTStoTM RPG routine for , 639*t*
- duration built-in functions for, 148, **148**
- duration codes for, 187, 211–212, 211*t*, 146, 146*t*, 296
- extract portion of (EXTRACT), 206*t*, 211, 212, 216*t*, 580*t*
- extract portion of (%SUBDT), 126*t*, 187, 211, 296, 580
- format codes for, 189*t*, 209–210, 210*t*, 211*t*, 327*t*, 631–641, 636*t*
- free-format syntax and, processing using, 579–582, **580**, **581**
- length of field for, 633–634
- MOVEL and , 207*t*, 209, 212, 271*t*, 281, 324–328, 324*t*, **325**, **326**, 327*t*, **328**, 344, 568, **569**, 570–571, 570*t*, **571**, 571*t*
- MtoDate RPG routine for, 640*t*
- retrieve from system (TIME), 396–397, 396*t*, 397*t*, **397**
- RPG xTools routines for, 639–641, 639–641*t*
- separator (edit character) for, 634–635
- special values (*HIVAL and *LOVAL) in 637–638, 638*t*
- storage of, 633–634
- subtract duration (SUBDUR), 208*t*, 211, 212, 218*t*, 385–386, 385*t*, **386**, 580*t*
- test result field of (TEST), 16, 208*t*, 209, 218*t*, 221*t*, 390–391, 391*t*, **391**, 580*t*
- TimeToTM RPG routine for, 639*t*, 639
- TIMFMT (separator selection) keyword and, 635–636, 636*t*
- TMtoDTS RPG routine for, 640*t*
- %TIME, 126*t*, 189, 189*t*, 572–573, **573**, 581, 582–583, **582**, **583**
- TIME, 208*t*, 396–397, 396*t*, 397*t*, **397**
- syntax diagram for, 218*t*
- timeouts, 353, 556
- timestamp data type, 12*t*. *See also* time
- add duration to (ADDUR), 231, 231*t*, **230**
- convert to (%TIMESTAMP), 126*t*, 190, 190*t*, 581, 582–583, **582**, **583**
- convert to characters (%CHAR), 121*t*, 137–138, **138**, 571–572, **571**, **572**, 582
- convert to time value (%TIME), 126*t*, 189, 189*t*, 572–573, **573**, 581, 582–583, **582**, **583**
- DATFMT keyword for, 634–635, 634*t*
- difference between (%DIFF), 122*t*, 146, 146*t*, **146**, 211, 580–582
- duration built-in functions for, 148, **148**
- duration codes for, 146, 146*t*, 187, 211–212, 211*t*, 296
- extract portion of (EXTRACT), 206*t*, 211, 212, 216*t*, 580*t*
- extract portion of (%SUBDT), 126*t*, 187, 211, 296, 580
- format codes for, 190*t*, 209–210, 210*t*, 211*t*, 211, 636–637
- free-format syntax and, processing using, 579–582, **580**, **581**
- length of field for, 633–634
- separator (edit character) for, 634–635
- special values (*HIVAL and *LOVAL) in 637–638, 638*t*
- subtract duration (SUBDUR), 208*t*, 211, 212, 218*t*, 385–386, 385*t*, **386**, 580*t*
- test result field of (TEST), 16, 208*t*, 209, 218*t*, 221*t*, 390–391, 391*t*, **391**, 580*t*
- TimeToTM RPG routine , 639*t*
- TIMFMT (separator selection) keyword, 43*t*, 53*t*, 69*t*, 635–636, 636*t*
- TIMSEP, 638
- /TITLE TEXT directive, 31*t*
- %TLOOKUP, 126*t*, 191
- TMtoDate RPG routine , 640*t*
- TMtoDTS RPG routine , 640*t*
- TO infix operator, 469
- TOFILE keyword, 69*t*
- tokens (symbols) in expressions, 108, 204*t*
- top-down design and development, 462–465, **463**, **464**
- CASE structures and, 476
- modular programming and, 484
- structured programming and, 462–465, **463**, **464**
- top-down logic, 455
- *TOTC, 29*t*
- *TOTL, 29*t*
- translate (XLATE), 208*t*, 219*t*, 407–409, 407*t*, **408**, **409**
- translation patterns (%XLATE), 127*t*, 196, **196**
- trash indicator, 16
- %TRIM, %TRIML, %TRIMR , 120, 126*t*, 192–193, **192–193**
- TRUNCNBR keyword, 44*t*, 44
- Create Bound Program (CRTBNDRPG) and, 616

U

- U1–U8 indicators, 15*t*
- %UCS2, 127*t*, 193
- UPDATE date variable, 99–100*t*
- UPDATE reserved word, 26*t*
- UDAY reserved word, 26*t*
- UMONTH reserved word, 26*t*
- unary operators, 106

- UNDEFINE conditional preprocessor directive, 33*t*
- UNIX, 429
- UNLOCK (Unlock File), 208*t*, 250, 263, 292, 398–399, 398*t*, **399**, 501
- cursor position in file after execution of, 543*t*
 - data areas and, 503–505, **504**, **505**
 - file processing using, 530*t*
 - free-format syntax in, 221*t*
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- syntax diagram for, 218*t*
- %UNS, 127*t*, 167, 194–195, **195**
- %UNSH, 127*t*, 167, 194–195, **195**
- unsigned integer data type, 12*t*, 56–58, 57*t*, 58*t*
- UPDATE (Update Record), 88, 208*t*, 250, 400–402, 400*t*, **401**, **402**, 557–558, **558**
- cursor position in file after execution of, 543*t*, **548**
 - fields, identifying for (%FIELDS), 156, **156**
 - file processing using, 530*t*
 - free-format syntax in, 221*t*
- READ and, 19, 123*t*, 128, 168–169, **169**, 207*t*, 217*t*, 221*t*, 250, 268, 293, 332, 333, 351–353, 351*t*, **352**, **353**, 529, 530*t*, 543, 543*t*, **547**, 556, **556**, 594–595, **594**, **594**, **595**, 623
- READE and, 207*t*, 221*t*, 250, 355–356, 355*t*, **356**, 530*t*, 543, 543*t*, **547**, 550
- READP and, 207*t*, 217*t*, 218*t*, 221*t*, 250, 268, 357–359, 357*t*, **358**, **359**, 530*t*, 543, 543*t*, **547**
- READPE and, 123*t*, 128, 168–169, **169**, 207*t*, 218*t*, 221*t*, 250, 360–361, 360*t*, **361**, 530*t*, 543, 544, 543*t*, **547**
- syntax diagram for, 219*t*
 - workstation device files and, 552*t*
- URL encoding, 600–601
- *USA, 141*t*, 189*t*, 210*t*, 211*t*, 318*t*, 327, 327*t*, 634*t*, 636*t*
- separator for, 635
- user entry procedure (UEP), 518
- user-defined indicator, 15*t*
- user-written functions. *See* procedures
- USRCTL keyword, 339
- USROPR keyword, 53*t*
- USRPRF keyword, 44*t*
- Create Bound Program (CRTBNDRPG) and, 615
- UYEAR reserved word, 26*t*
- ## V
- VALUE, 69*t*
- prototyped parameters and, 517*t*
 - parameters and, 114–115, **114**, **115**
- values, programs, 214
- variables, 431
- %ADDR (get address), 119, 120, 120*t*, 129–130, **130**
 - %CHAR, 121*t*, 137–138, **138**, 571–572, **571**, **572**, 582
 - environment, 592–593, 593*t*
 - get (QtmhPutEnv) in, 590–592, 591*t*, **591**, 588–590, 589*t*, **589**, **590**
 - EXPORT keyword and, 437–439, **438**, **439**
 - get/set substring of a value in (%SUBST), 109, 120, 126*t*, 128, 187–189, **188–189**, 288, 387
 - global, 431, 435–436
 - host, in data areas, 503
 - IMPORT keyword and, 437–439, **438**, **439**
 - indicator, 20
 - local, 431, 435–436
 - programs, 214
 - reset to initial value (RESET), 365–367, 365*t*, **367**
 - scope of, 435–436, **436**
 - set length of (%LEN), 119, 124*t*, 170, **170**
- VARYING keyword, 69*t*
- prototyped parameters and, 517*t*
- verification lists
- %CHECK, 121*t*, 138–139, **139**, 140, **140**
 - %CHECKR, 140, **140**
- versions of RPG, xxii
- views, 531
- ## W
- WATFOR/WATFIV, xix
- weak parameter module, parameter passing and, 489
- Web (CGI) programming interfaces, 585–602
- ATOLL(), C runtime library function in, 601, **602**
 - bindable APIs and, 587
 - CALLB and CALLP with, 587
 - CGI application programming interface (APIs) and, 587–600, 587*t*
 - CRTBNDRG and, 601
 - CRTRPGMOD and, 601

- environment variables and, 592–593, 593*t*
 escape characters and escape sequences in, 601
 in-line ILE procedures and, 648, **648**, **649**
 METHOD-GET/METHOD-POST and, 600–601
 PARM and, 207*t*, 217*t*, 209, 343–346, 343*t*, **345**, **346**,
 491–493, 588, 590–592, 591*t*, **591**, 594–595, 594*t*,
594, **595**
 QHTTSPVR library for, 588
 QtmhCvtDB (convert to database structure) in, 587*t*,
 596–600, 598–599*t*, **599**, **600**
 QtmhGetEnv (get value of environment variable) in,
 587*t*, 588–590, 589*t*, **589**, **590**
 QtmhPutEnv (put environment value) in, 587*t*, 590–592,
 591*t*, **591**
 QtmhRdSTin (read from standard input/stdin) in, 587*t*,
 594–595, 594*t*, **594**, **595**
 QtmhWrStout (write to standard input/stdout) in, 587*t*,
 595–596, 596*t*, **596**
 QZHB CGI service program for, 588
 QzhbCgiParse in, 587*t*
 QzhbCgiRecvState in, 587*t*
 QzhbCgiSendState in, 587*t*
 QzhbCgiUtils in, 587*t*
 SRVPGM*t* service program containing, 588
 URL encoding and, 600–601
 WebSphere Development Studio ILE RPG, xxiii
 WHEN, 105, 128, 208*t*, 402–403, 402*t*, **403**, 443, 455, 457,
 465, 467*t*
 AND and, 233, 233*t*, **233**
 built-in functions and, 119
 comparison and, 110–111, **110**, **111**
 conditioning and branching operations using, 451*t*
 conditioning indicators not supported by, 209
 fixed-format in, 220, 220*t*
 free-format syntax in, 221*t*, 566
 indicator use and, 444, 445
 natural expression support in, 220, 220*t*
 nesting in, 227
 OR and, 340, 340*t*, **340**
 OTHER and, 341, 341*t*, **341**
 SELECT and, 373–374, 373*t*, **373**
 syntax diagram for, 219*t*
 workstations
 acquire (ACQ), 205*t*, 221*t*, 215*t*, 228–229, 228*t*, **229**,
 332, 364, 530*t*, 552*t*
 device file processing for, 551–552, 552*t*
 display message on (DSPLY), 278–279, 278*t*, **279**
 dumb terminals as, 551
 force next input from (NEXT), 207*t*, 217*t*, 221*t*,
 332–333, 332*t*, **333**
 nonprogrammable (NPT), 551
 operations used with, 552*t*
 personal computers (PC) as, 552
 post device information to (POST), 207*t*, 217*t*, 221*t*,
 349–350, 349*t*, **350**, 530*t*, 552*t*, 555, **555**
 read next changed records (READC), 207*t*, 217*t*, 221*t*,
 354, 354*t*, **354**, 530*t*, 552*t*, 557, **557**
 release device file in (REL), 207*t*, 218*t*, 221*t*, 292, 530*t*, 552*t*
 timeouts, 353, 556
 write/read workstation file format (EXFMT), 206*t*, 216*t*,
 221*t*, 293, 293*t*, **293**, 530*t*, 552*t*, 554–555, **554**
 WORKSTN files, 84
 WRITE (Write to File), 88, 208*t*, 404–405, 404*t*, **405**, 529,
 558, **559**
 cursor position in file after execution of, 543*t*, **548**
 EXFMT and, 554–555, **554**
 file processing using, 530*t*
 free-format syntax in, 221*t*
 syntax diagram for, 219*t*
 workstation device files and, 552*t*
- ## X
- XFOOT (Cross Foot Array), 195, **195**, 208*t*, 406, 406*t*, **406**
 syntax diagram for, 219*t*
 %XFOOT, 127*t*
 XLATE (Translate), 208*t*, 407–409, 407*t*, **408**, **409**
 syntax diagram for, 219*t*
 %XLATE, 127*t*, 196, **196**
 XML, xxiii
 XML parser
 error codes in, 412, **412**
 exception/error codes and, 165–166*t*
 handler subprocedure for, 409–411, 415–418, 422–424, **422**
 handler subprocedure parameter definitions, for
 XML-SAX, 423*t*
 %HANDLER for, 123*t*, 128, 159–161, **160**, **161**,
 162–166, **163**, 165*t*, 409, 410, 415–418, **416**
 named constants in, 419
 processing XML data in, using XML-INTO, 417, **417**
 PSDS subfields in, 412, **412**, **419**
 receiver variable of XML-INTO and, 411–415
 %STATUS and, extended exception/error codes for,
 125*t*, 155, 182–184, **182**, 183*t*, 222, 224, 413–415*t*,
 420–422*t*
 %XML and, 127*t*, 128, 197–198, 198–200*t*, 409, 410
 XML-INTO, 409–418, 409*t*

XML parser, *continued*XML-SAX event codes and, 164*t*XML-SAX and, 418–424, 418*t*%XML, 127*t*, 128, 197–198, 198–200*t*, 409, 410XML-INTO, xxiii, 128, 409–418, 409*t*. *See also* XML parser; XML-SAXfree-format syntax in, 221*t*%HANDLER and, 123*t*, 128, 159–161, **160**, **161**,162–166, **163**, 165*t*, 409, 410, 415–418, **416**%XML and, 127*t*, 128, 197–198, 198–200*t*, 409, 410XML-SAX, xxiii, 128, 418–424, 418*t*. *See also* XML parser; XML-INTOevent codes for, 423–424*t*event codes generated by, 164*t*free-format syntax in, 221*t*%HANDLER and, 123*t*, 128, 159–161, **160**, **161**,162–166, **163**, 165*t*, 409, 410, 415–418, **416**XML parser exception/error codes and, 165–166*t*%XML and, 127*t*, 128, 197–198, 198–200*t*, 409, 410

XMLxx built-in function, xxiii

XOR, %BITXOR, 136–137, **136**, 137*t***Y**

*Y, 212

*YEAR, 26*t* 27*t*, 146*t*, 187, 211*t*, 212, 231, 293%YEARS , 127*t*, 148*YMD, 141*t*, 210, 210*t*, 318*t*, 327, 327*t*, 634*t*

separator for, 635

ZZ operation extender, 213*t*Z-ADD (Zero and Add), 26, 208*t*, 344 , 425, 425*t*, **425**syntax diagram for, 219*t*Z-SUB (Zero and Subtract), 208*t*, 426, 426*t*, **426**syntax diagram for, 219*t*

zero suppression, 93

control codes in, 94, 95*t*

*DAT and *UPDATE, 100

*ZERO, 23*t*zero, 95*t*

zeroing

zero and add (Z-ADD), 26, 208*t*, 219*t*, 344 , 425, 425*t*, **425**zero and subtract (Z-SUB), 208*t*, 219*t*, 426, 426*t*, **426***ZEROS, 23*t*zone-to-zone moves (MxxZO operations), 315, 315*t*zoned decimal data, 620 , 624–626, **624**, 625*t*

zoned fields, decimal data, 622

ZTools source codes, 643–650

