

Index

SYMBOLS

! (logical negation) operator, 106
 != (not equals) operator, 103, 334–335
 " (double quotes), 41
 # (hash) symbol, 130
 % (remainder) operator, 79–80
 && (AND) operator, 105
 flag enums, 317–319
 overloading, 337
 () (cast) operator, 339–340
 * (multiplication) operator, 79–80
 ++ (increment) operator, 88–91
 + (plus) operator, 78–79, 81–82
 -- (decrement) operator, 88–91
 - (delete) operator, 192
 - (minus) operator, 78–79
 - (subtraction) operator, 79–80
 / (division) operator, 79–80
 /// (three-forward-slash delimiter), 351
 ; (semicolons), statements without, 12–13
 < (less than) operator, 103
 <= (less than or equal to) operator, 103
 == (equals) operator, 103, 334–335
 = operator, 16
 > (greater than) operator, 103
 >= (greater than or equal to) operator, 103
 ? (conditional) operator, 107
 @ character, 43
 [] (index) operator, 426, 444–448
 \n (newline) character, 42
 \() (parenthesis) operator (delete slash),
 86–87

^ (exclusive OR) operator, 106
 {} (curly braces), 61
 code blocks, 99–101
 || (OR) operator, 104, 105
 flag enums, 317–319
 overloading, 337
 ~ (bitwise complement) operator, 113, 423

A

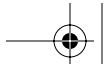
Abort() method, 557
 abstract classes, 263–268
 interfaces, comparing, 294
 abstract members, defining, 264–265
 access
 base members, 261
 code, 6
 code, security, 650
 fields, 196
 instance fields, 194–195
 interfaces, 611–612
 metadata, 511–512
 modifiers. *See* access modifiers
 property-backing fields, 226
 referent type members, 638–639
 scope, 101–102
 static fields, 213
 access modifiers, 202–204, 345
 classes, 344
 encapsulation, circumventing, 650
 private, 244–245
 properties, 228–229
 protected, 245–246



698  Index

- accessors, arrays, 65
- acronyms (C#), 660–661
- add blocks, 506
- adding
 - comments, 22–25
 - items (to Dictionary<TKey, TValue>), 426
 - operators, 335–336
- addresses, 630–639
 - unsafe code, 630–632
- aggregation, 247
 - single inheritance, 288–290
- aliases
 - handling, 529
 - namespace qualifiers, 348–349
 - using directives, 154–155
- allocation, stacks, 636
- alternate threads, passing data to/from, 595–600
- ambiguity (with this keyword), avoiding, 198–199
- AND (&&) operator, 105
 - flag enums, 317–319
 - overloading, 337
- anonymous methods, 475–479
 - outer variables, 479–481
- APIs (application programming interfaces)
 - encapsulation, 626–628
 - wrapping, 628–629
- app (application) domains, 652–653
- application programming interfaces. *See* APIs
- applications. *See also* programs
 - compiling, 3–4
 - limitations, running multiple times, 578–579
- applying
 - arrays, 65–71
 - classes, generic, 385–387
 - variables, 18
 - weak references, 356
- arbitrary state, passing, 603–605
- arguments, passing command-line, 156
- arithmetic operators, 79–80
- ArrayList class, 420–423
- arrays, 60–73
 - accessors, 65
 - applying, 65–71
 - assigning, 61–65
 - declaring, 60–61, 66
 - foreach loops, 438–439
 - instance methods, 70–71
 - instantiation, 61–65
 - jagged, 65
 - length, 67
 - methods, 68–70
 - parameters, 163–166
 - redimensioning, 70
 - strings as, 71–73
 - troubleshooting, 73, 74–75
- as operator, inheritance, 271–272
- assemblies, 3
 - attributes, 522–523
 - CIL, 4
 - CLI, 653–655
 - referencing, 341–346, 345–346
 - reflection. *See* reflection
 - targets, modifying, 342–343
- Assert() method, 85
- assigning
 - arrays, 61–65
 - pointers, 634–636
 - variables, 15, 16–17
- assignment operators, 87–88
- binary operators, combining, 337
- bitwise, 112–113
 - delegate events, 497
- association, data, 214
- associativity, operators, 81
- asynchronous results patterns, 592–606
- atomicity, 551, 567, 592
- attributes, 509
 - assemblies, 522–523
 - constructors, initializing, 526–530
 - customizing, 524–525
 - FlagsAttribute, 533–535
 - FlagsAttribute, enums, 319–320
 - IndexerNameAttribute, 446
 - MethodImplAttribute, 583–584
 - named parameters, 532–533
 - overview of, 520–524
 - predefined, 535
 - properties, decorating, 521
 - retrieving, 525–526
 - returning, 524
 - searching, 525–526
 - serialization, 538–546
 - serialization, customizing, 541
 - serialization, ISerializable interface, 541–542
 - serialization, versioning, 542
 - StructLayoutAttribute, 620–621
 - System.AttributeUsageAttribute, 531–533





- System.Collections.Hashtable, 531
- System.ConditionalAttribute, 535–537
- System.NonSerializable, 540
- System.ObsoleteAttribute, 537–538
- System.SerializableAttribute, 539–540
- System.SerializableAttribute, 546–547
- System.STAThreadAttribute, 613
- ThreadStaticAttribute, 581–583
- AttributeUsageAttribute, 535
- avoiding
 - ambiguity (this keyword), 198–199
 - deadlock, 576–577, 592
 - locking, 572–573, 577
 - unboxing, 311

B

- background worker patterns, 606–610
- backward compatibility, 545
- base classes
 - constraints, 401–402
 - overriding, 250–262
 - refactoring, 238
- Base Class Library. *See* BCL
- base constructors, invoking, 262
- base interfaces, declaring classes, 286–287
- base members, 261
- base types, casting, 241
- BCL (Base Class Library), 5, 6, 658
 - Boolean type, 38–39
 - integers, 31
- behavior
 - catch blocks, 181
 - System.Exception, 370
- benefits of generics, 388–389
- binary digits, 31
- binary operators, 79–80, 335–337
- BinarySearch() method, 68
- BinaryTreeString, foreach loops, 458
- BinaryTree<T> class, 412
- bits, 31, 108–109
- bitwise complement (~) operator, 113, 423
- bitwise operators, 108–113. *See also* operators
 - assignment, 112–113
 - for positions, 126
- blocks
 - add, 506
 - catch, 175
 - catch, behavior, 181

- catch, generic, 180–181
- catch, internals, 370
- checked, 377
- finally, 177
- general catch, 368–370
- remove, 506
- try, 175
- unchecked, 377–378
- unsafe code, 631
- unsafe code, designating for, 639–640
- Boolean expressions, 102–107
 - if statements, 96–97
- Boolean types, 38–39
 - numbers, converting to, 57
- boxing
 - runtime, 309
 - value types, 305–311
- break statements, 124–126
- BubbleSort() method, 466–469
- buffers, overruns, 68
- bytes, 108–109

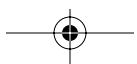
C

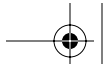
C

- pointer declaration, 633
- short data type, 31

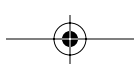
C#

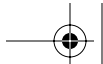
- casting, 243
- CIL and ILDASM, 25–28
- comments, 22–25
- common acronyms, 660–661
- compilation to machine code, 644–646
- compilers, 643–644
- console input/output, 18–22
- generics without, 380–384
- HelloWorld program, 2–6
- identifiers, 8–9
- keywords, 7–9
- Main method, declaring, 11–12
- methods, 10–11
- overview of, 1–2
- preprocessor directives, 130–136
- properties, 46–48
- statements, 12
- statements, delimiters, 12
- statements, without semicolons, 12–13
- syntax, 7–14
- type definition, 9–10
- variables, 14–18
- whitespace, 13–14



**700**  **Index**

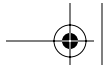
- C++
 - arrays, declaring, 61
 - Boolean expression coding errors, 103
 - buffer overruns, 68
 - default parameters, 149
 - (delete) operator, 192
 - deterministic destruction, 358, 648
 - dispatch method calls, 255
 - global methods, 148
 - global variables, 211
 - header files, 151
 - implicit overriding, 252
 - multiple inheritance, 246
 - operator-only statements, 79
 - pointer declaration, 633
 - preprocessing, 130
 - pure virtual functions, 266
 - short data type, 31
 - strings, concatenation at compile time, 43
 - structs, defining types, 303
 - switch statements, 124
 - templates, 399
- callers, 141
- calling
 - binary operators, 336
 - constructors, 205–206
 - dispatch methods, 255
 - external functions, 626–628
 - methods, 140–146
- call sites, 157–158
- call stacks, 157–158
- camel casing, 10
- CAS (code access security), 517
- case sensitivity, 3, 11
- casting, 53
 - implicit casts, 241
 - inheritance, 241–242
 - inheritance, defining custom conversions, 243–244
 - interfaces, 284–285
 - methods, generics, 413
 - types, 241
 - types, conversion without, 58–59
- cast (()) operator, 54, 57, 339–340
- catch blocks, 175
 - behavior, 181
 - generic, 180–181
 - internals, 370
- catching
 - exceptions, 174–175, 177–178, 367–368
 - exceptions, general catch blocks, 368–370
 - categories of data types, 50–53
 - categories of types, 298
 - centralizing initialization, 210–211
 - chaining multicast delegates, 492
- characters
 - @, 43
 - escape sequences, 40
 - newline (n), 42
 - operators, 82–83
 - types, 39
 - Unicode, 39–41
 - whitespace, 13–14
- checked blocks, 55, 377
- checked conversions, 54–55
 - exceptions, 376–378
- CIL (Common Intermediate Language),
 - 4, 642, 656
 - code, boxing, 306
 - delegate events, 504
 - generic representations, 414–415
 - ILDASM, 25–28
 - properties, 223–224
 - System.SerializableAttribute, 546–547
- classes, 185–189
 - abstract, 263–268
 - abstract, comparing interfaces and, 294
 - access modifiers, 202–204, 344
 - assemblies, availability outside of, 343–344
 - base, constraints, 401–402
 - base, overriding, 250–262
 - base, refactoring, 238
 - base interfaces, declaring, 286–287
 - BCL, 658
 - BinaryTree<T>, 412
 - casting, implementing, 284
 - collections, 420–433
 - collections, ArrayList, 420–423
 - collections, interfaces, 433–443
 - collections, List<T>, 420–423
 - CommandLineInfo, 516
 - const modifiers, 218–219
 - constructors, 204–211
 - data association, 214
 - defining, 189–192
 - definition, 9
 - deriving, 239–240
 - encapsulation, 187
 - encapsulation, information hiding, 202–204
 - encapsulation, objects group data with
 - methods, 192
 - Exception, inheritance, 178





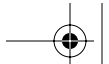
- generics, 385–387
- generics, defining, 387–388
- generics, determining support, 519–520
- hierarchies, 188
- inheritance, 187–189. *See also* inheritance
- inner, Java, 234
- instances, fields, 193–195
- instances, methods, 195–196
- instantiation, 189–192
- interfaces, comparing, 293
- Java, naming, 4
- libraries, 4, 341–342, 342
- members, 192
- members, private, 203
- nested, 232–234
- object-oriented programming, 186–187
- partial, 234–236
- polymorphism, 189
- properties, 220–232
- Queue, 432–433
- readonly modifiers, 218–219
- sealed, 250
- Stack, 431–432
- Stack<T>, 387
- static, 216–217
- static keyword, 211–218
- structs, 458–459
- System.Array, 69. *See also* arrays
- System.Net.WebClient, 606
- System.Threading.Interlocked, 573–575
- System.Threading.Mutex, 578
- this keyword, 196–202
- cleanup, resources, 356–363
- Clear() method, 68
- CLI (Common Language Infrastructure),
 - 1, 5, 641–642
 - app domains, 652–653
 - assemblies, 653–655
 - BCL, 658
 - C# compilation to machine code, 644–646
 - CIL, 656
 - CLS, 657–658
 - CTS, 656–657
 - defining, 642–643
 - implementing, 643–644
 - manifests, 653–655
 - metadata, 658–659
 - modules, 653–655
 - runtime, 647–652
 - runtime, code access security, 650–651
 - runtime, garbage collection, 647–648
 - runtime, garbage collection on .NET, 648–649
 - runtime, performance, 651–652
 - runtime, platform portability, 650–651
 - runtime, type safety, 649
 - types, 30
- Clone() method, 70
- Close() method, 201
- CLR (Common Language Runtime), 647
- CLS (Common Language Specification),
 - 5, 657–658
 - types, 30
- CLU language, 448–449
- clusters, 448–449
- code
 - access security, 6
 - blocks, statements, 99–101
 - boxing, 306
 - conventions, delegate events, 500–501
 - event-coding patterns, 499
 - excluding/including, 131–132
 - machine, 4, 642
 - machine, C# compilation to, 644–646
 - managed, 4
 - multithreading. *See* multithreading
 - paths, 149
 - P/Invoke, 616–630
 - pseudocode, 566
 - security, 650
 - source code listings, 667–691
 - unmanaged, 5
 - unsafe. *See* unsafe code
 - versioning, 537–538
- code access security (CAS), 517
- collections, 419–420
- classes, 420–433
- classes, ArrayList, 420–423
- classes, interfaces, 433–443
- classes, List<T>, 420–423
- dictionaries, 425–429
- empty/null, returning, 448
- index (I |) operators, 444–448
- iterators, 448–463
- linked lists, 433
- modifying, 443
- queues, 432–433
- sorted, 429–431
- stacks, 431–432
- COM
 - DLL registration, 655
 - Interop, 613





- command-line arguments, passing, 156
- CommandLineInfo class, 516
- comments
 - C#, 22–25
 - types, 23–24
- Common Intermediate Language. *See* CIL
- Common Language Infrastructure.
 - See* CLI
- Common Language Runtime (CLR), 647
- Common Language Specification. *See* CLS
- Common Type System. *See* CTS
- Compare() function, 436
- comparing
 - classes and interfaces, 293
 - enum switches and integers, 311–312
- comparison operators, 334–335
- compatibility
 - backward, 545
 - types, between enums, 315
 - types, generics, 396
- compilers
 - C#, 643–644
 - JIT, 645
 - Mono, 3, 664–665
 - .NET, 663–664
- compiling
 - applications, 3–4
 - CIL code, 4
- components, 645
- composite formatting, 21
- conditional expressions, 102. *See also* Boolean expressions
- conditional logical operators, 337
- conditional (?) operator, 107
- configuring compilers, .NET, 664
- console
 - executable assembly types, 342
 - input/output, 18–22
- constants
 - expressions (const), 92
 - math, 101
- const modifiers, 218–219
- constraints
 - base classes, 401–402
 - constructors, 404
 - generics, 396–409
 - inheritance, 405
 - interfaces, 400–401
 - limitations of, 405–409
 - multiple, 403
 - specifying, 412
 - structs, 402–403
- constructors
 - attributes, initializing, 526–530
 - classes, 204–211
 - constraints, 404
 - declaring, 205–206
 - default, 206–207
 - defining, 392
 - finalizers, 207
 - inheritance, 262
 - initialization, centralizing, 210–211
 - initializers, 209
 - overloading, 208
 - static, 215–216
- constructs, associating XML comments, 350–351
- Contains() method, 432
- contextual keywords, 8
- continue statements, 127–128
- contravariance, 243
- conventions
 - code, delegate events, 500–501
 - naming, properties, 224–225
- conversions
 - as operator, 271–272
 - checked, 54–55
 - custom, defining, 243–244
 - data types, 53–60
 - between enums and strings, 315–316
 - exceptions, 376–378
 - implicit, 340
 - numbers to Boolean types, 57
 - numeric with TryParse() methods, 183–184
 - operators, 339
 - operators, guidelines, 340–341
 - types without casting, 58–59
 - unchecked, 54–55
- copying, avoiding, 311
- covariance, 243
- CTS (Common Type System), 5, 656–657
- curly braces ({}), 61
 - code blocks, 99–101
- customization
 - attributes, 524–525
 - attribute serialization, 541
 - conversions, defining, 243–244
 - delegate events, 505–506
 - exceptions, 373–378

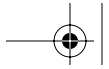




D

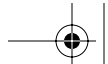
- data association, 214
- data persistence to files, 200–201
- data retrieval from files, 201–202
- data types, 15–16
 - arrays, 60–73
 - Boolean, 38–39
 - categories of, 50–53
 - characters, 39
 - conversion, 53–60
 - conversion, without casting, 58–59
 - decimal, 32–33
 - delegates, 468–469
 - floating-point, 31–33
 - hexadecimal notation, 36–37
 - integers, 30–31
 - literal values, 33–35
 - null, 49–50
 - nullable modifiers, 53
 - numeric types, 30–38
 - overview of, 29–30
 - parameters, P/Invoke, 618–619
 - round-trip formatting, 37–38
 - strings, 41–45
 - System.Text.StringBuilder, 48
 - TryParse() method, 59–60
 - void, 50
- deadlock, 551–552
 - avoiding, 576–577, 592
- decimal types, 32–33
- declaring
 - arrays, 60–61, 66
 - constraints, base classes, 401–402
 - constraints, interfaces, 400–401
 - constructors, 205–206
 - delegates, data types, 470
 - delegates, method returns, 496
 - events, delegates, 498–500
 - explicit members, 286
 - fields, as readonly, 219
 - generic delegate types, 502
 - getter/setter methods, 220
 - GreaterThanHandler types, 471
 - instance fields, 193
 - interfaces, generics, 389–390
 - jagged arrays, 66
 - Main method, 11–12
 - methods, 146–151
 - nullable value types, 384
 - pointers, 632–634
 - properties, 221–223
 - static classes, 216–217
 - static constructors, 215–216
 - static fields, assigning at, 213
 - static properties, 225
 - two-dimensional arrays, 61
 - types, aliases, 154–155
 - types, from unmanaged structs, 620–621
 - variables, 15, 16
 - variables, of class types, 190
 - VirtualAllocEx() APIs, 618
 - volatile fields, 573
- decorating properties with attributes, 521
- Decrement() method, 566
- decrement (--) operator, 88–91
- default constructors, 206–207
- default() operator, 63
- default operators, value types, 304
- default parameters, 149
- default values, 393–394
- defining
 - abstract classes, 263–264
 - abstract members, 264–265
 - cast (()) operators, 339
 - classes, 9, 189–192
 - classes, generics, 387–388
 - CLI, 642–643
 - constructors, 392
 - custom attributes, 524–525
 - custom conversions, 243–244
 - custom exceptions, 373–378
 - delegate types, 470–471
 - enums, 312–313, 318–319
 - finalizers, 357–358, 392
 - index (I) operators, 444–445
 - iterators, 449
 - methods, generics, 410
 - methods, subscriber, 482–484
 - namespace, 346–349
 - nested classes, 232–234
 - objects, 190–191
 - partial classes, 235
 - preprocessor symbols, 132–133
 - properties, 221–222, 229
 - publishers, 484–485
 - read-only properties, 227–228
 - serializable exceptions, 375–376
 - structs, 300–301
 - types, 9–10
 - warnings, 133





- delegates, 465–466
 - anonymous methods, 475–479
 - data types, 468–469
 - events, 496–506
 - events, coding conventions, 500–501
 - events, customizing, 505–506
 - events, declaring, 498–500
 - events, encapsulating publications, 498
 - events, encapsulating subscriptions, 497–498
 - events, generics, 501–503
 - events, internals, 503–505
 - functions, mapping pointers to, 629–630
 - instantiating, 471–475
 - internals, 469–470
 - invoking, 486–487
 - invoking, exception sequences, 494
 - invoking, instances with a return, 496
 - invoking, sequential invocation, 490–491
 - method pointers, 466–481
 - methods, returns, 495–496
 - multicast (and observer patterns), 482–496
 - operators, 488–490
 - outer variables, 479–481
 - pass-by-references, 495–496
 - passing, 472–473
 - types, defining, 470–471
 - delete (-) operator, 192
 - delimited comments, 23
 - delimiters
 - statements, 12
 - three-forward-slash (///), 351
 - Dequeue() method, 432
 - dereferencing pointers, 636–638
 - deriving
 - classes, 239–240
 - inheritance, 238–250
 - interfaces, 292–293
 - types, 189
 - types, casting, 241
 - deserialization, 543–544. *See also* serialization
 - design, synchronization, 576–577
 - deterministic destruction, 648
 - C++, 358
 - deterministic finalization, using statement, 359–361
 - diagramming interfaces, 290–291
 - dictionaries, collections, 425–429
 - Dictionary<TKey, TValue>, 425–429
 - dimensions, retrieving size, 70–71
 - directives
 - import, wildcards in, 152
 - Imports (Visual Basic.NET), 152
 - preprocessor (C#), 130–136
 - using, 151–155
 - using, aliasing, 154–155
 - using, nested, 153–154
 - dispatch method calls, 255
 - Dispose() method, 9
 - division (/) operator, 79–80
 - DLL (Dynamic Link Library), 4
 - documentation, XML, 352–353
 - documents, saving, 539–540
 - domains, app, 652–653
 - double quotes ("), 41
 - do/while loops, 113–115
 - Dynamic Link Library. *See* DLL
- ## E
- editors, hints for visual, 135–136
 - emitting errors and warnings, 133
 - empty catch block internals, 370
 - empty collections, returning, 448
 - encapsulation
 - access modifiers, circumventing, 650
 - APIs, 626–628
 - classes, 187
 - classes, information hiding, 202–204
 - classes, objects group data with
 - methods, 192
 - publications, 498
 - subscriptions, 497–498
 - types, 343–343
 - EndInvoke() method, 602
 - Enqueue() method, 432
 - enums, 311–320
 - defining, 312–313, 318–319
 - as flags, 316–319
 - FlagsAttribute, 319–320
 - strings, converting, 315–316
 - type compatibility, 315
 - values, 312
 - equality (==) operator, 103
 - equality operators, 103–104
 - Equals() method, overriding, 326–334
 - equals (==) operator, 334–335
 - errors
 - arrays, 73, 74–75
 - emitting, 133



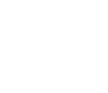


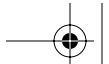
- handling, exceptions, 171–184
- handling, IEnumerator<T> interfaces, 442–443
- handling, multicast delegates, 493–495
- handling, P/Invoke, 621–623
- handling, Win32, 621–623
- infinite recursion, 168–169
- reporting, 181–183
- trapping, 173–178
- escape sequences, 39
- event keyword, 499
- events, 465–466
 - delegates, 496–506
 - delegates, coding conventions, 500–501
 - delegates, customizing, 505–506
 - delegates, declaring, 498–500
 - delegates, encapsulating publications, 498
 - delegates, encapsulating subscriptions, 497–498
 - delegates, generics, 501–503
 - delegates, internals, 503–505
 - notification, multithreading, 575–576
 - resetting, 579–581
- Exception class, inheritance, 178
- exceptions
 - behavior of System.Exception, 370
 - catching, 174–175, 177–178, 367–368
 - checked/unchecked conversions, 376–378
 - customizing, 373–378
 - error handling with, 171–184
 - general catch blocks, 368–370
 - handling, 365
 - handling, background worker patterns, 610
 - handling, guidelines, 371–373
 - multiple exception types, 365–367
 - NullReferenceException, 487
 - rethrowing, 182–183
 - serializable, 375–376
 - specifiers, 368
 - subscribers, handling, 494
 - throwing, 67, 173, 181–182, 366, 493
 - types, 179
 - unhandled, threads, 563–564
- excluding code, 131–132
- exclusive OR (^) operator, 106
- execution
 - managed, 5
 - time, 5
 - VES, 642

- explicit cast, 54
- explicit members
 - declaring, 286
 - implementing, 281–282
- exponential notation, 35
- expressions
 - Boolean, 102–107
 - Boolean, if statements, 96–97
 - constant (const), 92
 - typeof, 512
- Extensible Markup Language. *See* XML
- external functions
 - calling, 626–628
 - declaring, 617
- extern alias directives, 348–349

F

- false operator, overloading, 339
- features, 2–3
- Fibonacci numbers, 114
- Fibonacci series, 114
- fields
 - access, 196
 - declaring as readonly, 219
 - instances, 193–195, 212–214
 - property-backing, accessing, 226
 - static, 212–214
 - this keyword, 196–197
 - virtual, properties as, 229–232
 - volatile, 573
- FIFO (first in, first out), collections, 432
- files
 - data persistence to, 200–201
 - data retrieval from, 201–202
 - header, 151
 - loading, 200–201
 - XML, documentation, 352–353
- finalization
 - garbage collection, 362
 - guidelines, 363
 - using statement, 359–361
- finalizers, 357–358
 - constructors, 207
 - defining, 392
- finally blocks, 177
- FindAll() method, 424–425
- firing event notification, 501, 575
- first in, first out (FIFO), collections, 432
- fixed statements, 634





flags, enums as, 316–319
 FlagsAttribute, 533–535
 enums, 319–320
 floating-point types, 31–33
 inequality, 83–86
 operators, 83
 float types, overflowing, 86
 flow control, 113
 foreach loops, 119–120
 if statements, 96–97
 for loops, 116–119
 nested if statements, 97–99
 overview of, 92–96
 switch statements, 122–124
 while and do/while loops, 113–115
 foreach loops, 119–120
 arrays, 438–439
 BinaryTreeString, 458
 IEnumerable<T> interfaces, 439–443
 foreach statements, iterator state, 453
 for loops, 116–119
 format items, 21
 format strings, 21
 formatting
 composite, 21
 custom exceptions, 373–378
 numbers, hexadecimal notation, 36–37
 round-trip, 37–38
 whitespace, 13–14
 forms, Windows Forms, 611–613
 f-reachable queues, 362
 functions. *See also* methods
 Compare(), 436
 external, calling, 626–628
 external, declaring, 617
 pointers, mapping, 629–630
 pure virtual, 266

G

garbage collection, 6
 CLI runtime, 647–648
 .NET, 648–649
 well-formed types, 353–356
 garbage collectors, 191
 general catch blocks, 368–370
 generics, 379
 benefits, 388–389
 catch blocks, 180–181
 classes, 385–387
 classes, defining, 387–388

constraints, 396–409
 C# without, 380–384
 delegate events, 501–503
 instantiating, 415–418
 interfaces with structs, 389–391
 internals, 413–418
 Java, 417
 methods, 409–413
 methods, casting, 413
 methods, specifying constraints, 412
 nested types, 395–396
 reflection, 517, 535–536
 reflection, determining support, 519–520
 types, 385–396
 GetHashCode() method, overriding, 324–326
 GetInvocationList() method, 495
 GetLength() method, 70
 getter methods, 220
 access modifiers, 228–229
 GetType() member, 511–512
 global methods, 148
 global variables, 211
 C++, 211
 goto statements, 128–130
 GreaterThanHandler type, 474–475
 declaring, 471
 greater than (>) operator, 103
 greater than or equal to (>=) operator, 103
 grouping statements, 140
 guest computers, 616

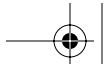
H

handlers, add/remove, 506
 hardcoding values, 33–35
 hash (#) symbol, 130
 Hashtable class, 425–429
 header files, 151
 heaps, 51
 reference types, 299
 HelloWorld program, 2–6
 CIL output, 26–27
 hexadecimal notation, 36–37
 hierarchies, classes, 188
 hints for visual editors, 135–136

I

ICollection<T> interface, 438–443
 IComparable<T> interface, 435–437
 identifiers, C#, 8–9





- IDictionary<TKey, TValue> interfaces, 434–435
- IDisposable interfaces, 9
 - P/Invoke, 625–626
 - resource cleanup, 359–360
- IEnumerable<T> interfaces
 - foreach loops, 439–443
- if statements
 - flow control, 96–97
 - nested, 97–99
- ILDASM, 25–28
- IList<T> interface, 434–435
- immutable strings, 18, 47
 - modifying, 636–637
- implementation
 - classes, casting, 284
 - CLI, 643–644
 - comparison operators, 334–335
 - Equals() method, 330–333
 - explicit members, 281–282
 - GetHashCode() method, 325–326
 - IComparable<T> interface, 436
 - implicit members, 282–283
 - interfaces, 276–278, 279–284
 - interfaces, generics, 390
 - interfaces, multiple times on single classes, 391–392
- implicit casts, 54, 57, 241
- implicit conversions, 340
- implicit members, implementing, 282–283
- implicit overriding, 252
- import directives, wildcards in, 152
- Imports directive (Visual Basic.NET), 152
- including code, 131–132
- increment (++) operator, 88–91
- indentation, 14. *See also* whitespace
- IndexerNameAttribute, 446
- index ([]) operator, 426
 - collections, 444–448
- inequality (!=) operator, 103
- inequality operators, 83–86. *See also* operators
- inferencing types, 411–412
- infinite recursion errors, 168–169
- information hiding, 202–204
- inheritance
 - abstract classes, 263–268
 - base classes, overriding, 250–262
 - base members, 261
 - casting, 241–242
 - casting, defining custom conversions, 243–244
 - classes, 187–189
 - constraints, 405
 - constructors, 262
 - defined terms, 237–238
 - deriving, 238–250
 - Exception class, 178
 - interfaces, 285–287
 - interfaces, multiple, 287–291
 - is operator, 270–271
 - methods, 240
 - multiple, 246
 - new modifier, 256–260
 - as operator, 271–272
 - private access modifiers, 244–245
 - protected access modifiers, 245–246
 - sealed modifiers, 260–261
 - single, 246–249
 - single, aggregation, 288–290
 - System.Object, 268–270
 - types, 189
 - value types, 304–305
- initialization
 - attributes through constructors, 526–530
 - centralizing, 210–211
 - jagged arrays, 65
 - structs, 302–303
 - three-dimensional arrays, 64–65
 - two-dimensional arrays, 66
- initializers, constructors, 209
- inner classes, Java, 234
- input, 18–22
- instances
 - array methods, 70–71
 - fields, 193–195, 212–214
 - methods, 195–196
 - Type, 512
- instantiation, 12
 - arrays, 61–65
 - classes, 189–192
 - delegates, 471–475
 - generics, 415–418
 - threads, 554–555
- integers, 30–31
 - enum switches, comparing, 311–312
 - values, overflowing, 55–56, 376
- interfaces, 273
 - access, 611–612
 - APIs. *See* APIs





interfaces (*Continued*)

- casting, 284–285
 - classes, comparing, 293
 - collection classes, 433–443
 - constraints, 400–401
 - deriving, 292–293
 - diagramming, 290–291
 - generics with structs, 389–391
 - ICollection<T>, 438–443
 - IComparable<T>, 435–437
 - IDictionary<TKey, TValue>, 434–435
 - IDisposable, P/Invoke, 625–626
 - IDisposable, resource cleanup, 359–360
 - IEnumerable<T>, foreach loops, 439–443
 - IList<T>, 434–435
 - implementing, 276–278, 279–284
 - inheritance, 285–287
 - inheritance, multiple, 287–291
 - ISerializable, 541–542
 - polymorphism, 275–279
 - value types, 304–305
 - versioning, 291–293
- internals
- anonymous methods, 478–479
 - delegate events, 503–505
 - delegates, 469–470
 - empty catch block, 370
 - generics, 413–418
 - multicast delegates, 492–493
 - outer variables, 480–481
 - properties, 223–223
- interoperability
- languages, 6
 - platforms. *See* platforms
- invalid referent types, 633
- Invoke() method, accessing interfaces, 611–612
- invoking
- asynchronous results patterns, 595–596
 - base constructors, 262
 - delegates, 486–487
 - delegates, exception sequences, 494
 - delegates, instances with a return, 496
 - delegates, sequential invocation, 490–491
 - members, reflection, 513–517
 - P/Invoke, 616–630
- ISerializable interface, 541–542
- is operator, inheritance, 270–271
- items
- adding, to Dictionary<TKey, TValue>, 426
 - format, 21

iteration

- dictionary classes, 428
 - state, sharing, 441
- iterators
- collections, 448–463
 - defining, 449
 - examples of, 455–456
 - multiple, 461–462
 - overview of, 460–461
 - state, 453–454
 - syntax, 450–451
 - values, yielding from, 451–453
 - yield statements, break, 459
 - yield statements, characteristics of, 462–463
 - yield statements, return, 456–458

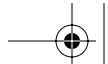
J

- jagged arrays, 65
- declaring, 66
- Java
- arrays, declaring, 61
 - classes, naming, 4
 - exception specifiers, 368
 - generics, 417
 - implicit overriding, 252
 - inner classes, 234
 - virtual methods, 251
 - wildcards in import directives, 152
- JIT (just-in-time) compiler, 645
- jitting, 4, 645
- Join() method, 556
- jump statements, 124–130
- just-in-time (JIT) compiler, 645

K

- keywords, 3
- C#, 7–9
 - checked/unchecked conversions, 54–55
 - event, 499
 - lock, 570–571
 - new, assigning arrays, 62
 - new, instantiating objects, 191
 - static, 211–218
 - this, 196–202
 - this, avoiding locking, 572–573
 - this, calling constructors, 209–210
 - throw, 366
 - yielding, 451





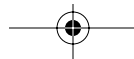
L

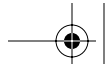
language interoperability, 6
last in, first out (LIFO), 380
 collections, 431
length
 arrays, 67
 strings, 46–47
Length member, 67
less than (<) operator, 103
less than or equal to (<=) operator, 103
libraries, 4
 BCL, 5, 6, 658
 classes, 341–342, 342
LIFO (last in, first out), 380
 collections, 431
limitations
 applications, running multiple times,
 578–579
 of constraints, 405–409
line-based statements, 12
lines, specifying numbers, 135
Liskov, Barbara, 448
lists
 collections, 420
 linked lists, 433
 parameters, 148
 parameters, passing variables, 164
List<T> class, 420–423
literals, 41–44
 verbatim string, 42
literal values, 33–35
loading files, 200–201
localized applications, building, 39
local storage, threads, 582
local variables, 15
 multithreading, 567–568
locking, avoiding, 577
lock keyword, 570–571
lock statements, value types, 308–309
logical negation (!) operator, 106
logical operators, 104–106, 110–112
loops
 for, 116–119
 decrement (--) operators, 88–91
 do/while, 113–115
 foreach, 119–120
 foreach, arrays, 438–439
 foreach, BinaryTreestring, 458
 foreach, IEnumerable<T> interfaces, 439–443

while, 113–115, 440
yield return statements, 457

M

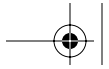
machine code, 4, 642
 C# compilation to, 644–646
Main method, 10, 566
 declaring, 11–12
Main() method, 155–158
 multiple, 157
managed code, 4
managed execution, 5
managing multithreading, 555–557
manifests, CLI, 653–655
mapping function pointers, 629–630
math constants, 101
members
 abstract, defining, 264–265
 base, 261
 classes, 192
 classes, private, 203
 explicit, declaring, 286
 explicit, implementing, 281–282
 GetType(), 511–512
 implicit, implementing, 282–283
 Length, 67
 Object, overriding, 323–334
 pointers, accessing referent types, 638–639
 Rank, 70
 reflection, invoking, 513–517
 sealing, 260
 System.Object, 269
messages, turning off warnings, 134
metadata, 24, 658–659
 accessing, 511–512
 reflection. *See also* reflection
MethodImplAttribute, 583–584
methods
 Abort(), 557
 anonymous, 475–479
 anonymous, outer variables, 479–481
 array, instances, 70–71
 arrays, 68–70
 Assert(), 85
 BinarySearch(), 68
 BubbleSort(), 466–469
 C#, 10–11
 calling, 140–146
 Clear(), 68



**710** ■ **Index**methods (*Continued*)

- Clone(), 70
- Close(), 201
- Contains(), 432
- declaring, 146–151
- Decrement(), 566
- delegates, returns, 495–496
- Dequeue(), 432
- dispatch calls, 255
- Dispose(), 9
- EndInvoke(), 602
- Enqueue(), 432
- Equals(), overriding, 326–334
- external, declaring, 617
- FindAll(), 424–425
- generics, 409–413
- generics, casting, 413
- generics, specifying constraints, 412
- GetHashCode(), overriding, 324–326
- GetInvocationList(), 495
- GetLength(), 70
- getter, 220
- getter, access modifiers, 228–229
- global, 148
- inheritance, 240
- initialization, 210–211
- instances, 195–196
- Invoke(), accessing interfaces, 611–612
- Join(), 556
- Main(), 10, 155–158, 566
- Main(), declaring, 11–12
- Main(), multiple, 157
- MoveNext(), 440
- names, 145
- namespaces, 142–144
- naming, 11
- overloading, 169–171
- overview of, 139–140
- parameters, 145
- parameters, declaring, 148–149
- Parse() method, 320
- Peek(), 432
- pointers, delegates, 466–481
- Pop(), 380, 432
- Push(), 380
- recursion, 166–169
- refactoring, 148
- ReferenceEquals(), 327–330
- returning, 145–146
- returning, declaring, 149–151
- Reverse(), 68
- scope, 144–145
- setter, 220
- setter, access modifiers, 228–229
- Sort(), 68
- statements, calling, 146
- static, 215
- Store(), 200
- strings, 44–45
- subscribers, defining, 482–484
- System.Console.Read(), 19–20
- System.Console.ReadKey(), 20
- System.Console.ReadLine(), 18
- System.Console.Write(), 20
- System.Console.WriteLine(), 20–21
- threads, starting in separate, 552–553
- Thread.Sleep(), 556–557
- ThreadStart, 555
- ToString(), 320, 324
- TryParse(), 59–60
- TryParse(), numeric conversion with, 183–184
- types, names, 144
- unsafe code, 631
- virtual, 251
- Microsoft IL (MSIL). *See* CIL
- Microsoft Visual Basic. *See* Visual Basic
- Microsoft Visual Basic.NET, Imports directive, 152
- Microsoft Visual Studio.NET, 137
- Miller, J., 5
- minus (-) operator, 78–79
- modification
 - assembly targets, 342–343
 - collections, 443
 - immutable strings, 636–637
 - variables, values, 17
- modifiers
 - access, 202–204, 345
 - access, circumventing encapsulation, 650
 - access, classes, 344
 - access, private, 244–245
 - access, properties, 228–229
 - access, protected, 245–246
 - const, 218–219
 - new, 256–260
 - nullable, 53
 - readonly, 218–219
 - sealed, 260–261
 - unsafe, 631
 - virtual, 251–256





- modules, 342
 - CLI, 653–655
- monitoring
 - threads, 568–569
 - threads, state, 592
- Mono compiler, 3, 664–665
- MoveNext() method, 440
- MSIL (Microsoft IL). *See* CIL
- multicast delegates (and observer patterns), 482–496
- multiple constraints, 403
- multiple exception types, 365–367
- multiple inheritance, 246
- multiple interface inheritance, 287–291
- multiple iterators, 461–462
- multiple Main() methods, 157
- multiple statements, 13
- multiple threads, 567–568
- multiple type parameters, 394–395
- multiplication (*) operator, 79–80
- multithreading, 549
 - atomicity, 551
 - deadlock, 551–552
 - local variables, 567–568
 - managing, 555–557
 - overview of, 550–552
 - patterns, 591–592
 - patterns, asynchronous results, 592–606
 - patterns, background worker, 606–610
 - patterns, Windows Forms, 611–613
 - pooling, 561–563
 - running, 552–557
 - starting, 553–554
 - synchronization, 565–584
 - synchronization, design, 576–577
 - synchronization, event notification, 575–576
 - synchronization, locking, 570–575
 - synchronization, MethodImplAttribute, 583–584
 - synchronization, resetting events, 579–581
 - synchronization,
 - System.Threading.Interlocked, 573–575
 - synchronization,
 - System.Threading.Monitor, 568–569
 - synchronization, ThreadStaticAttribute, 581–583
 - synchronization, types, 578–584
 - threads, passing parameters to, 557–561
 - ThreadStart methods, 555

- timers, 584–590
- uncertainty, 552
- unhandled exceptions, 563–565

N

- named parameters, attributes, 532–533
- names
 - classes, Java, 4
 - conventions, properties, 224–225
 - conventions, type parameters, 389
 - methods, 11, 145
 - parameters, 158
 - scope, 101–102
 - types, 144
- namespaces
 - alias qualifiers, 348–349
 - defining, 346–349
 - methods, 142–144
 - nesting, 347
- nested classes, 232–234
- nested generic types, 395–396
- nested if statements, 97–99
- nested using directives, 153–154
- nesting
 - delegate data types, 471
 - namespaces, 347
- .NET
 - compilers, 663–664
 - garbage collection, 648–649
 - garbage collection in, 354–355
 - NGEN tool, 645
- new keyword
 - arrays, assigning, 62
 - objects, instantiating, 191
- newline (n) character, 42
- new modifiers, 256–260
- new operators, value types, 303
- NGEN tool (.NET), 645
- notation
 - exponential, 35
 - hexadecimal, 36–37
- not equals (!=) operator, 334–335
- notification
 - asynchronous results patterns, 597
 - events, 501
 - events, multithreading, 575–576
 - threads, completion of, 600–602
- nowarn:<warn list> option, 134
- nullable modifiers, 53





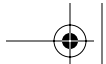
nullable value types, 383–384
 null collections, returning, 448
 NullReferenceException, 487
 null references, checking for, 487–488
 null types, 49–50
 numbers
 Boolean types, converting, 57
 Fibonacci, 114
 hexadecimal notation, 36–37
 lines, specifying, 135
 numeric conversion with TryParse()
 methods, 183–184
 numeric types, 30–38
 decimal, 32–33
 floating-point, 31–33
 hexadecimal notation, 36–37
 integers, 30–31
 literal values, 33–35
 round-trip formatting, 37–38
 TryParse() method, 59–60

O

Object members, overriding, 323–334
 object-oriented programming, classes, 186–187
 objects
 data association, 214
 defining, 190–191
 identity, 327
 lock, selecting, 571–572
 resurrecting, 363–363
 observer patterns, multicast delegates, 482–496
 operands, 78
 operational polymorphism, 169
 operators, 77–92
 =, 16
 AND (&&), 105
 AND (&&), flag enums, 317–31
 AND (&&), overloading, 337
 adding, 335–336
 arithmetic, 79–80
 as, inheritance, 271–272
 assignment, 87–88
 assignment, bitwise, 112–113
 assignment, combining with binary, 337
 assignment, delegate events, 497
 associativity, 81
 binary, 335–337
 bitwise, 108–113
 bitwise, complement (~), 113, 423

bitwise, for positions, 126
 Boolean expressions, 102–107
 cast (()), 54, 57, 339–340
 characters, 82–83
 comparison, 334–335
 conditional (?), 107
 conditional logical, 337
 constant expressions (const), 92
 conversion, 339
 conversion, guidelines, 340–341
 decrement (--), 88–91
 default(), 63
 default, value types, 304
 delegates, 488–490
 delete (-), 192
 division (/), 79–80
 equality, 103–104
 equals (==), 103, 334–335
 exclusive OR (^), 106
 false, overloading, 339
 floating-point types, 83
 greater than (>), 103
 greater than or equal to (>=), 103
 increment (++), 88–91
 index ([]), 426
 index ([]), collections, 444–448
 inequality, 83–86
 inequality (!=), 103
 is, inheritance, 270–271
 less than (<), 103
 less than or equal to (<=), 103
 logical, 104–106, 110–112
 logical negation (!), 106
 minus (-), 78–79
 multiplication (*), 79–80
 new, value types, 303
 not equals (!=), 334–335
 OR (| |), 104, 105
 OR (| |), flag enums, 317–319
 OR (| |), overloading, 33
 overloading, 334–341
 parenthesis / (/), 86–87
 plus (+), 78–79
 plus (+), applying with strings, 81–82
 postfix, 90
 precedence, 81, 137138
 prefix, 90
 relational, 103–104
 remainder (%), 79–80
 shift, 109–110





- subtraction (-), 79–80
- true, overloading, 339
- unary, 78–79, 338–339
- options, nowarn:warn list, 134
- order of precedence, operators, 81, 137–138
- OR (| |) operator, 104, 105
 - flag enums, 317–319
 - overloading, 33
- outer variables, 479–481
- out parameter values, 232
- output, 18–22
 - CIL for HelloWorld, 26–27
 - parameters, 161–163
- overflowing
 - float type, 86
 - integer values, 55–56, 376
- overloading
 - constructors, 208
 - methods, 169–171
 - operators, 334–341
 - operators, AND (&&), 337
 - operators, false, 339
 - operators, OR (| |), 337
 - operators, unary, 338–339
 - true operators, 339
- overriding
 - base classes, 250–262
 - Equals() method, 326–334
 - GetHashCode() method, 324–326
 - implicit, 252
 - Object members, 323–334
 - properties, 251–252
 - ToString() method, 324
- overruns, buffers, 68

P

- parameters, 141, 158–166
 - arrays, 163–166
 - covariance/contravariance, 243
 - data types, P/Invoke, 618–619
 - default, 149
 - lists, 148
 - Main() method, 155–158
 - methods, 145
 - methods, declaring, 148–149
 - named, attributes, 532–533
 - names, 158
 - output, 161–163
 - out values, 232

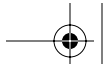
- overview of, 139–140
- references, 160–161
- ref values, 232
- threads, passing to, 557–561
- types, multiple, 394–395
- types, naming conventions, 389
- values, 158–159
- variables, defining index ([] operators, 447–448
- parenthesis () operator, 86–87
- Parse() method, 320
- partial classes, 234–236
- Pascal casing, 10
- pass-by-references, 495–496
- passing
 - anonymous methods, 476
 - arbitrary state, 603–605
 - command-line arguments, 156
 - data to/from alternate threads, 595–600
 - data to/from threads, 592
 - delegates, 472–473
 - method returns, 145–146
 - parameters to threads, 557–561
 - this keyword, 199–200
 - variables, by references, 160
 - variables, out only, 161
 - variables, parameter lists, 164
- paths, code, 149
- patterns
 - event-coding, 499
 - multithreading, 591–592
 - multithreading, asynchronous results, 592–606
 - multithreading, background worker, 606–610
 - multithreading, Windows Forms, 611–613
 - observer, multicast delegates, 482–496
- Peek() method, 432
- performance, CLI runtime, 651–652
- permissions, CAS, 517
- P/Invoke, 616–630
 - APIs, wrapping, 628–629
 - error handling, 621–623
 - external functions, calling, 626–628
 - external functions, declaring, 617
 - functions, mapping pointers, 629–630
 - IDisposable interfaces, 625–626
 - parameter data types, 618–619
 - ref or out, applying, 619–620
 - SafeHandle, 623–624
 - StructLayoutAttribute, 620–621





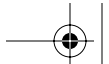
- placeholders, swapping, 21
 - platforms, 615
 - P/Invoke, 616–630
 - P/Invoke, applying ref or out, 619–620
 - P/Invoke, calling external functions, 626–628
 - P/Invoke, declaring external functions, 617
 - P/Invoke, error handling, 621–623
 - P/Invoke, IDisposable interfaces, 625–626
 - P/Invoke, mapping function pointers, 629–630
 - P/Invoke, parameter data types, 618–619
 - P/Invoke, SafeHandle, 623–624
 - P/Invoke, StructLayoutAttribute, 620–621
 - P/Invoke, wrapping APIs, 628–629
 - portability, 6, 650–651
 - plus (+) operator, 78–79
 - strings, applying with, 81–82
 - pointers, 630–639
 - assigning, 634–636
 - declaring, 632–634
 - dereferencing, 636–638
 - functions, mapping, 629–630
 - members, accessing referent types, 638–639
 - methods, delegates, 466–481
 - unsafe code, 630–632
 - polymorphism, 266–268
 - classes, 189
 - interfaces, 275–279
 - operational, 169
 - pooling threads, 561–563, 592
 - Pop() method, 380, 432
 - portability, platforms, 6, 650–651
 - postfix operators, 90
 - precedence, operators, 81, 137–138
 - predefined attributes, 535
 - predefined types, 29
 - prefix operators, 90
 - preprocessor directives (C#), 130–136
 - primitives, 29
 - priority, increasing, 556
 - private access modifiers, 244–245
 - private members, 203
 - programs
 - HelloWorld, 2–6
 - HelloWorld, CIL output, 26–27
 - Undo, 380–381
 - properties
 - access modifiers, 228–229
 - attributes, decorating, 521
 - C#, 46–48
 - CIL code, 223–224
 - classes, 220–232
 - declaring, 221–223
 - defining, 221–222, 229
 - internals, 223–223
 - naming conventions, 224–225
 - out/ref parameter values, not allowed as, 232
 - overriding, 251–252
 - read-only, 46, 227–228
 - static, 225
 - thread.IsBackground, 556
 - ThreadState, 556
 - validation, 225–227
 - virtual fields, 229–232
 - write-only, 227–228
 - protected access modifiers, 245–246
 - pseudocode, 566
 - publications, encapsulation, 498
 - publishers
 - defining, 484–485
 - subscribers, combining, 485–486
 - publish-subscribe patterns. *See* observer patterns
 - punctuation, 2
 - pure virtual functions, 266
 - Push() method, 380
- ## Q
- qualifiers, namespace alias, 348–349
 - Queue class, 432–433
 - queues
 - collections, 432–433
 - f-reachable, 362
 - Queue<T> class, 432–433
- ## R
- race conditions, 567
 - Ragsdale, S., 5
 - Rank member, 70
 - RCW (runtime callable wrapper), 613
 - readonly modifiers, 218–219
 - read-only properties, 46, 227–228
 - recursion, 166–169
 - infinite recursion errors, 168–169
 - redimensioning arrays, 70
 - refactoring
 - base classes, 238
 - methods, 148
 - ReferenceEquals() method, 327–330





- references
 - assemblies, 341–346, 345–346
 - null, checking for, 487–488
 - parameters, 160–161
 - pass-by-references, 495–496
 - strong, 355–356
 - types, 51–53, 159–160, 299
 - types, instantiating generics, 416–418
 - weak, 355–356
 - reflection, 509
 - generics, 517, 535–536
 - generics, determining support, 519–520
 - members, invoking, 513–517
 - overview of, 510–520
 - System.Type, accessing metadata, 511–512
 - type parameters, determining type of, 518–519
 - type parameters, obtaining, 520
 - ref parameter values, 232
 - registering
 - subscribers, 486
 - unhandled exceptions, 563–564
 - relational operators, 103–104
 - remainder (%) operator, 79–80
 - remove blocks, 506
 - reporting errors, 181–183
 - reserved words. *See* keywords
 - resetting events, 579–581
 - resources
 - cleanup, 356–363
 - SafeHandle, 623–624
 - restrictions,
 - System.AttributeUsageAttribute, 531–533
 - resurrecting objects, 363–363
 - rethrowing exceptions, 182–183
 - retrieving attributes, 525–526
 - returning
 - attributes, 524
 - empty/null collections, 448
 - Main() method, 155–158
 - methods, 145–146
 - methods, declaring, 149–151
 - methods, delegates, 495–496
 - values, 17
 - return statements, 150
 - return values, 141
 - Reverse() method, 68
 - reversing strings, 72–73
 - round-trip formatting, 37–38
 - running
 - applications, 3–4
 - multithreading, 552–557
 - runtime, 5, 642
 - arrays, defining size at, 63
 - boxing, 309
 - CLI, 647–652
 - CLI, code access security, 650–651
 - CLI, garbage collection, 647–648
 - CLI, garbage collection on .NET, 648–649
 - CLI, performance, 651–652
 - CLI, platform portability, 650–651
 - CLI, type safety, 649
 - CLR, 647
 - virtual methods, calling, 253
 - runtime callable wrapper (RCW), 613
- ## S
- SafeHandle, P/Invoke, 623–624
 - safety (type), 6, 649
 - saving documents, 539–540
 - scope, 101–102
 - methods, 144–145
 - sealed classes, 250
 - sealed modifiers, 260–261
 - searching attributes, 525–526
 - security, code, 6, 650
 - selecting objects, lock, 571–572
 - semicolons (;), statements without, 12–13
 - sequences, escape, 39
 - sequential delegate invocation, 490–491
 - serialization
 - attributes, 538–546
 - attributes, customizing, 541
 - attributes, ISerializable interface, 541–542
 - attributes, versioning, 542
 - exceptions, 375–376
 - series, Fibonacci, 114
 - setter methods, 220
 - access modifiers, 228–229
 - sharing state, 441
 - shift operators, 109–110
 - simple assignment operator, 16
 - single inheritance, 246–249
 - aggregation, 288–290
 - single-line comments, 23
 - sites, call, 157–158
 - size
 - arrays, defining at runtime, 63
 - dimensions, retrieving, 70–71

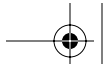




716 Index

- sorted collections, 429–431
- SortedDictionary<TKey, TValue>, 429–431
- SortedList<T> class, 429–431
- sorting IComparable<T> interfaces, 436
- Sort() method, 68
- source code listings, 667–691
- source languages, 656
- specializing types, 189
- specifiers, exceptions, 368
- specifying
 - constraints, 412
 - default values, 393–394
 - multiple constraints, 403
 - types, parameters, 411
- splitting statements, 13
- Stack class, 431–432
- stacks, 51
 - allocating, 636
 - call, 157–158
 - collections, 431–432
 - unwinding, 158
- Stack<T> class, 387, 431–432
- starting
 - methods, in separate threads, 552–553
 - threads, 553–554
- state
 - arbitrary, passing, 603–605
 - iterators, 453–454
 - sharing, 441
 - threads, monitoring, 592
 - unsynchronized, 565–567
- statements
 - break, 124–126
 - C#, 12
 - code blocks, 99–101
 - continue, 127–128
 - fixed, 634
 - flow control, 93–96
 - foreach, iterator state, 453
 - goto, 128–130
 - grouping, 140
 - if, 96–97
 - if, nested, 97–99
 - jump, 124–130
 - line-based, 12
 - lock, value types, 308–309
 - method calls, 146
 - operator-only, 79
 - Redim (Visual Basic), 70
 - return, 150
 - without semicolons, 12–13
- switch, 12, 121, 122–124
- throw, reporting errors, 181–183
- using, 9, 361
 - using, deterministic finalization, 359–361
 - yield, characteristics of iterators, 462–463
 - yield break, iterators, 459
 - yield return, iterators, 456–458
- static classes, 216–217
- static constructors, 215–216
- static fields, 212–214
- static keyword, 211–218
- static methods, 215
 - strings, 44–45
- static properties, 225
- Store() method, 200
- strings
 - as arrays, 71–73
 - concatenation at compile time (C++), 43
 - enums, converting, 315–316
 - format, 21
 - immutable, 18, 47
 - immutable, modifying, 636–637
 - length, 46–47
 - literals, 41–44
 - methods, 44–45
 - properties, 46–48
 - reversing, 72–73
 - types, 41–45
 - verbatim literals, 42
- strong references, 355–356
- StructLayoutAttribute, 620–621
- structs
 - classes, 458–459
 - constraints, 402–403
 - defining, 300–301
 - generics with interfaces, 389–391
 - initializing, 302–303
 - unmanaged, declaring types from, 620–621
 - value types, 298–305
- subscribers
 - exceptions, handling, 494
 - methods, defining, 482–484
 - publishers, combining, 485–486
- subscriptions, encapsulation, 497–498
- subtraction (-) operator, 79–80
- subtypes, 189
- suffixes, literal values, 35
- swapping
 - data between positions in arrays, 66



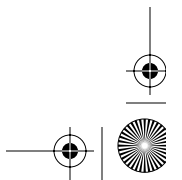


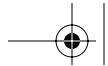
placeholders, 21
 switch statements, 12, 121, 122–124
 symbols. *See also* characters; operators
 hash (#), 130
 preprocessor, defining, 132–133
 synchronization, 592
 multithreading, design, 576–577
 multithreading, event notification, 575–576
 multithreading, locking, 570–575
 multithreading, MethodImplAttribute, 583–584
 multithreading, resetting events, 579–581
 multithreading,
 System.Threading.Interlocked, 573–575
 multithreading, System.Threading.Monitor, 568–569
 multithreading, ThreadStaticAttribute, 581–583
 multithreading, types, 578–584
 threads, 565–584
 syntax
 C#, 7–14
 checked blocks, 55
 iterators, 450–451
 System.Array class, 69. *See also* arrays
 System.AttributeUsageAttribute, 531–533
 System.Collections.Hashtable attributes, 531
 System.ConditionalAttribute, 535–537
 System.Console.ReadKey() method, 20
 System.Console.ReadLine() method, 18
 System.Console.Read() method, 19–20
 System.Console.WriteLine() method, 20–21
 System.Console.Write() method, 20
 System.Delegate, 470. *See also* delegates
 System.Exception. *See also* exceptions
 behavior of, 370
 customizing, 374
 System.Net.WebClient class, 606
 System.NonSerializable attribute, 540
 System.Object, inheritance, 268–270
 System.ObsoleteAttribute, 537–538
 System.Runtime.Serialization.OptionalFieldAttribute, 545
 System.SerializableAttribute, 539–540
 CIL, 546–547
 System.STAThreadAttribute, 613
 System.Text.StringBuilder data type, 48
 System.Threading.AutoResetEvent, 579–581
 System.Threading.Interlocked class, 573–575
 System.Threading.ManualResetEvent, 579–581

System.Threading.Monitor, 568–569
 System.Threading.Mutex, 578
 System.Threading.ParameterizedThreadStart
 delegate type, 557–558
 System.Threading.ThreadPool, 561–563
 System.Threading.ThreadStart, 554–554
 System.Timers.Timer, 587–590
 System.Type, accessing metadata, 511–512

T

targets, modifying assemblies, 342–343
 templates, C++, 399
 this keyword, 196–202
 constructors, calling, 209–210
 locking, avoiding, 572–573
 thread.IsBackground property, 556
 threads
 alternate, passing data to/from, 595–600
 completion, notification of, 600–602
 creating (in C# 2.0), 554–555
 instantiating, 554–555
 local storage, 582
 local variables, 567–568
 managing, 555–557
 methods, starting in separate, 552–553
 monitoring, 568–569
 multithreading. *See* multithreading
 parameters, passing to, 557–561
 pooling, 561–563, 592
 starting, 553–554
 state, monitoring, 592
 synchronization, 565–584
 thread safe conditions, 567
 unhandled exceptions, 563–564
 thread-safe, incrementing/decrementing, 91–92
 Thread.Sleep() method, 556–557
 ThreadStart methods, 555
 ThreadState property, 556
 ThreadStaticAttribute, 581–583
 three-dimensional arrays, initializing, 64–65
 three-forward-slash delimiter (///), 351
 throwing exceptions, 67, 173, 181–182, 366, 493
 throw keyword, 366
 throw statements, reporting errors, 181–183
 timers, multithreading, 584–590
 tools, NGEN (.NET), 645
 ToString() method, 320, 324
 trapping errors, 173–178



**718**  **Index**

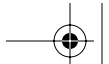
troubleshooting
 arrays, 73, 74–75
 error handling with exceptions,
 171–184
 infinite recursion, 168–169
true operator, overloading, 339
try blocks, 175
TryParse() method, 59–60
 numeric conversion with, 183–184
two-dimensional arrays
 declaring, 61
 initializing, 66
Type instances, 512
typeof type
 avoiding locking, 572–573
 expression, 512
types
 aliases, declaring, 154–155
 arrays, 60–73
 Boolean, 38–39
 casting, 241
 categories of, 50–53, 298
 characters, 39
 checking, 649
 comments, 23–24
 compatibility, between enums, 315
 compatibility, generics, 396
 conversion, 53–60
 conversion, without casting, 58–59
 CTS, 656–657
 data, 15–16. *See also* data types
 decimal, 32–33
 definition, 9–10
 delegates, defining, 470–471
 derived, 189
 encapsulation, 343–343
 exceptions, 179
 exceptions, catching, 367
 float, overflowing, 86
 floating-point, 31–33
 floating-point, inequality, 83–86
 floating-point, operators, 83
 GetType() member, 511–512
 GreaterThanHandler, 474–475
 GreaterThanHandler, declaring, 471
 hexadecimal notation, 36–37
 inferencing, 411–412
 inheritance, 189
 invalid referent, 633

is operators, 270–271
literal values, 33–35
multiple exception, 365–367
names, 144
nested generics, 395–396
null, 49–50
nullable modifiers, 53
numeric, 30–38
parameters, multiple, 394–395
parameters, naming conventions, 389
references, 51–53, 159–160, 299
references, instantiating generics,
 416–418
round-trip formatting, 37–38
safety, 6, 649
specializing, 189
strings, 41–45
structs. *See* structs
subtypes, 189
synchronization, 578–584
TryParse() method, 59–60
typeof, avoiding locking, 572–573
typeof, expression, 512
values, 50–51, 159–160. *See also* value
 types
values, instantiating generics, 415–416
values, nullable, 383–384
variables, declaring, 190
void, 50
well-formed. *See* well-formed types

U

unary operators, 78–79, 338–339
unboxing, 308. *See also* boxing
 avoiding, 311
uncertainty (multithreading), 552
unchecked blocks, 56, 377–378
unchecked conversions, 54–55
 exceptions, 376–378
Undo programs, 380–381
unhandled exceptions, 173
 threads, 563–564
Unicode characters, 39–41
unmanaged code, 5
unmanaged structs, declaring types from,
 620–621
unsafe code, 615, 630–632
 blocks, designating, 639–640





- immutable strings, modifying, 637–638
- unsafe modifier, 631
- unsynchronized state, 565–566
- unwinding stacks, 158
- using directives, 151–155
 - aliasing, 154–155
 - nested, 153–154
- using statements, 9, 361
 - deterministic finalization, 359–361

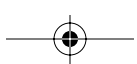
V

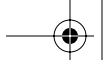
- validating properties, 225–227
- values
 - default, 393–394
 - enums, 312
 - integers, overflowing, 55–56, 376
 - iterators, yielding from, 451–453
 - literal, 33–35
 - parameters, 158–159
 - return, 17, 141
- value types, 50–51, 159–160, 297
 - boxing, 305–311
 - default operators, 304
 - enums, 311–320
 - generics, instantiating, 415–416
 - inheritance, 304–305
 - interfaces, 304–305
 - lock statements, 308–309
 - new operators, 303
 - nullable, 383–384
 - structs, 298–305
 - structs, initializing, 302–303
- variables
 - applying, 18
 - C#, 14–18
 - classes, declaring types, 190
 - declaring, 15, 16
 - local, multithreading, 567–568
 - outer, 479–481
 - out only, passing, 161
 - parameters, defining index ([]) operators, 447–448
 - parameters, passing lists, 164
 - references, passing by, 160
 - scope, 101–102
 - values, modifying, 17
- verbatim string literals, 42
- versioning
 - code, 537–538

- interfaces, 291–293
 - serialization, 542
- VES (Virtual Execution System), 4, 642
- VirtualAllocEx() API, 618
- virtual computers, 616
- Virtual Execution System. *See* VES
- virtual field properties, 229–232
- virtual functions, pure, 266
- virtual methods, 251
- virtual modifiers, 251–256
- Visual Basic
 - arrays, redimensioning, 70
 - classes, accessing instances, 197
 - default parameters, 149
 - global methods, 148
 - global variables, 211
 - line-based statements, 12
- Visual Basic.NET, Imports directive, 152
- visual editors, hints of, 135–136
- Visual Studio.NET, 137
- void types, 50
- volatile fields, 573

W

- warnings
 - emitting, 133
 - messages, turning off, 134
- weak references, 355–356
- well-formed types, 323
 - assemblies, referencing, 341–346
 - encapsulation, 343–343
 - garbage collection, 353–356
 - namespace, defining, 346–349
 - Object members, overriding, 323–334
 - operators, overloading, 334–341
 - resource cleanup, 356–363
 - unary operators, 338–339
 - XML comments, 349–353
- while loops, 113–115, 440
- whitespace, C#, 13–14
- wildcards in import directives, 152
- Windows
 - executables, 342
 - Forms, 611–613
- Win32 error handling, 621–623
- wrappers
 - APIs, 628–629
 - RCW, 613
- write-only properties, 227–228





X

XML (Extensible Markup Language)
comments, 349–353
delimited comments, 24
documentation, 352–353
overview of, 24–25
single-line comments, 24

Y

yield break statements, iterators, 459
yielding keywords, 451
yield return statements, iterators, 456–458
yield statements, iterators, 462–463

