interpolation, 47
interpreted queries, 398-404
  AsEnumerable operator, 403
  combining local queries with, 402
  execution, 401
  mechanism of operation, 400-400
interprocess communication (IPC), 648
Intersect operator, 457
into keyword, 393
IOrderedEnumerable, 452
IOrderedQueryable, 452
IPAddress class, 690
IPEndPoint class, 690
IProducerConsumerCollection<T>, 959
IProgress<T>, 627
IPv4/IPv6 addresses, 689
IQueryable<T>, 398-402
IReadOnlyCollection<T>, 326
IReadOnlyList<T>, 326
is operator, 110
IsAccessible method, 1039
ISerializable, 751-754
IStructuralComparable, 366
IStructuralEquatable, 366
ISymbol interface, 1037-1038
iteration statements, 77-79
  for loops, 78
  foreach loops, 79
  while and do-while loops, 77
iteration variables, 168
iterators, 181-184, 320
  composing sequences, 183
  semantics, 182-183
  try/catch/finally blocks, 183
  yield break statement, 182
IUnknown interface, 990
IV (Initialization Vector), 872
IXmlSerializable, 728, 736-738

## J

jagged arrays, 51
JIT (Just-In-Time) compilation, 3
Join method (strings), 248
Join method (threading), 578
Join operator (LINQ), 442-449
  basics, 443
  joining in fluent syntax, 445
  joining on multiple keys, 444
joining, 442-449
  SelectMany, 438
  strings, 248
JSON, 516-522
  JsonDocument, 519-522
  Utf8JsonReader, 516-518
  Utf8JsonWriter, 518
JsonCommentHandling, 518
JsonConverterAttribute, 743
JsonDocument, 519-522
  LINQ and, 521
  making updates with a JSON writer, 522
  reading JSON arrays, 520
  reading JSON objects, 520
  reading simple values, 520
JsonExtensionDataAttribute, 742
JsonIgnoreAttribute, 741
JsonPropertyNameAttribute, 742
JsonSerializationOptions, 744-747
JsonSerializer, 738-747
  basics, 727, 738
  controlling serialization with attributes, 741-743
  customizing data conversion, 743
  JsonSerializationOptions, 744-747
  serializing child objects, 739-740
  serializing collections, 740
JsonWriterOptions, 518
jump statements, 79-81
  break statement, 79
  continue statement, 80
  goto statement, 80
  return statement, 80
  throw statement, 81
Just-In-Time (JIT) compilation, 3

## K

key management, 875
keywords, 25-26
KnownFolders class, 678

## L

lambda expressions, 2, 165-169
  anonymous methods versus, 169
  asynchronous, 616
  captured variables and, 584

capturing iteration variables, 168
capturing outer variables, 166-169
composing, 374-376
element typing and, 375
explicitly specifying parameter types, 166
Func signatures and, 375
local methods versus, 169
subqueries, 388-391
lambda operator (=>), 165
language binding, 210
Large Object Heap (LOH), 538
Last operator, 460
Last-In First-Out (LIFO), 341
LastIndexOf method, 247
LastNode function, 477
late binding, 810
lazy execution (see deferred execution)
lazy initialization, 912-914
Lazy<T> class, 913
LazyInitializer class, 913
lazy quantifiers, 1003
LazyInitializer class, 913
left-associative operators, 66
let keyword, 397
LIFO (Last-In First-Out), 341
LIKE operator, 427
LinkedList<T>, 338
LINQ operators, 421-468
aggregation methods, 462-466
conversion methods, 457-460
element operators, 460-461
filtering, 425-429
generation methods, 467
grouping, 452-455
joining, 442-449
ordering, 450-452
overview, 422-425
projecting, 429-441
quantifiers, 466
sequence to element or value, 424
sequence to sequence, 423-424
set operators, 456
void to sequence, 425
Zip operator, 449
LINQ queries, 369-420
anonymous types, 396
basics, 369-371
composition strategies, 392-395
deferred execution, 382-388
fluent syntax, 371-378
interpreted queries, 398-404
into keyword, 393
JsonDocument and, 521
let keyword, 397
object initializers, 395
progressive query building, 392-393
projecting into an X-DOM, 497-500
projection strategies, 395-397
query expressions, 378-382
subqueries, 388-391
wrapping queries, 394
LINQ to XML, 469-500
annotations, 496
architectural overview, 469
documents and declarations, 487-491
LINQ to XML DOM, 470
names and namespaces, 491-496
X-DOM (see XML DOM)
XML declarations, 489
LINQ, JsonDocument and, 521
LINQ-to-objects queries, 370
list-like collections, 335-344
BitArray, 342
HashSet<T> and SortedSet<T>, 342
LinkedList<T>, 338
List<T> and ArrayList, 336-338
Queue<T> and Queue, 340
Stack<T> and Stack, 341
List<T> class, 325, 336-338
ListDictionary, 349
Listeners, 556-558
literals, 26
Load(byte[]) method, 790
LoadFile method, 790
LoadFrom method, 790
LoadFromAssemblyName method, 779
local methods, 93
in C# 7, 14
lambda expressions versus, 169
static local methods, 93
local queries, 402
local sequences, 369
local variables, 22
dynamic code generation, 826
var keyword, 64
LocalApplicationData, 673
localization, 286

lock statement, 81, 883
locking
  exclusive (see exclusive locking)
  immutable objects, 895
  locking around thread-safe objects, 892
  nonexclusive (see nonexclusive locking)
  read-only thread safety, 893
  static members, 892
  thread safety and, 582, 890-895
  thread safety and .NET Core types, 891-893
  thread safety in application servers, 893
lockTaken overloads, 884
logical negation operator (!), 41, 552
LOH (Large Object Heap), 538
LongCount operator, 462
LookupSymbols, 1040
loop iteration index, 942

M

Main method, 22
man-in-the-middle attack, 876
managed memory leaks, 542-545
ManualResetEvent, 906
ManualResetEventSlim, 906
MatchEvaluator delegate, 1008
Math class, 288
Max operator, 462
member overload resolution, 854-860
member types, 807
MemberInfo subclass, 807
memory barrier, 886
memory leaks
  diagnosing, 545
  managed memory leaks, 542-545
  timers and, 543
memory management, 3
  mapping structs to unmanaged memory, 984-988
  shared memory, 982-984
memory, stack-allocated/unmanaged, 972
memory-mapped files, 683-686
  (see also shared memory)
  cross-platform interprocess shared memory, 684
  random file I/O and, 683
  shared memory (Windows), 684
  working with view accessors, 685
Memory<T>, 969-970
MemoryStream, 647, 872
metacharacters, 1000
method group, 150
method overloading, 115
method parameters, 811-813
MethodAttributes.HideBySig, 835
methods, 2, 92-93
  emitting, 833-835
  expression-bodied, 92
  generic, 137
  local methods, 93
  overloading, 92
  overriding, 835
  pass-by-value versus pass-by-reference parameters, 93
  purpose of, 22
Microsoft Azure, 875
Microsoft Dataflow, 923
Min operator, 462
minus sign (-)
  negative infinity (−∞), 41
  negative zero (−0), 41
  removing delegate instances, 151
  subtraction operator, 39
mocking handler, 699
modules, assembly, 760, 818
Monitor.Enter, 884
Monitor.Exit, 884
Mono, 5
MoveNext method, 182
multicast delegates, 151
multidimensional arrays, 50
multiline comments (/* */), 27, 226
multiple dispatch, 857
MultipleActiveResultSets (MARS), 416
multithreaded program, 576
multithreaded timers, 919-920
multithreading (see parallel programming)
Mutex, 889

N

naked type constraint, 141
name hiding, 84

name scoping, 83
named arguments, 62
named groups, 1007
named pipes, 648-650
nameof operator, 106
namespace alias qualification (::), 87
namespaces, 23, 81-87
  advanced features, 85-87
  alias qualifiers, 87
  aliasing types and, 85
  attributes, 492
  extension methods and, 194
  extern aliases, 85
  name hiding, 84
  name scoping, 83
  nested using directives withing, 85
  prefixes, 492
  repeated, 84
  rules within, 83-85
  using directive and, 82
  using static directive and, 83
  X-DOM, 493-495
  XML, 491-496
  XmlReader and, 508
  XmlWriter and, 510
navigation properties, 412-414
  adding and removing entities from navigation collections, 413
  lazy loading, 414
  loading, 413
negative lookahead, 1004
negative lookbehind, 1004
nested locking, 887
nested types, 134-135
  obtaining, 799
  type names, 800
.NET Core
  about, 5
  CLR and BCL, 235-239
  collections (see collections)
  framework fundamentals, 243-313
  framework overview, 231-242
  .NET Core 3 new features, 232
  standard disposal semantics, 524
.NET Framework
  about, 5
  application frameworks, 239-242
  compilation, 24
.NET standard, 231-234
  .NET Framework and .NET Core compatibility, 234
  older standards, 234
  reference assemblies, 235
.NET standard 2.0, 233
.NET standard 2.1, 233
networking, 687-723
  addresses and ports, 689
  architecture, 687
  client-side classes, 692-706
  concurrency with TCP, 719
  receiving POP3 mail with TCP, 720-722
  sending mail with SmtpClient, 716
  TCP in UWP, 722-723
  URIs, 690-692
  using DNS, 715
  using FTP, 713-715
  using TCP, 717-720
  working with HTTP, 706-710
  writing an HTTP server, 710-713
new keyword, 113
Nodes function, 477
nonexclusive locking, 896-903
  lock recursion, 902
  reader/writer locks, 898-903
  semaphore, 896-898
  upgradeable locks, 901-903
nongeneric interfaces, 319
nonpublic constructors, 95
nonpublic members, 813
null coalescing operator (??), 69, 189
null operators, 69-70, 189
null strings, 246
null value, literal for, 34
null-coalescing assignment operators, 69
null-conditional operator (?.), 69, 161, 189
null-forgiving operator (!), 191
nullable annotation context, 192
#nullable enable directive, 192
nullable reference types, 191-193
  about, 11
  null-forgiving operator, 191
  separating annotation and warning contexts, 192
  treating nullable warnings as errors, 193
nullable types, 34
nullable value types, 185-190

alternatives to, 190
bool? with & and | operators, 188
boxing/unboxing nullable values, 186
implicit/explicit nullable conversions, 186
null operators and, 189
Nullable<T> struct, 185
operator lifting, 186
scenarios for, 189
Nullable<T> struct, 185
NullReferenceException, 178, 191
NumberFormatInfo, 273
NumberStyles, 274, 278
numeric format strings, 276-276
numeric literals, 36
in C# 7, 13
numeric suffixes, 37
type inference, 37
numeric suffixes, 37
numeric types, 36-43, 288-291
arithmetic operators, 39
BigInteger struct, 289
char type conversions, 46
Complex struct, 290
conversions, 38, 288
double versus decimal, 42
8- and 16-bit integral types, 41
increment and decrement operators, 39
Math class, 288
numeric literals, 36
Random class, 290
real number rounding errors, 43
special float and double values, 41
specialized operations on integral types, 39
unification, 853

## O

object initializers, 97
LINQ queries and, 395
optional parameters versus, 98
object instantiation, 115
object orientation, 1
object tracking, 410
object type, 116-120
boxing and unboxing, 117
GetType method and typeof operator, 118
object member listing, 119
static/runtime type checking, 118
ToString method, 119
Object...
Object.Equals method, 298-299
object.Equals static method, 299, 304
object.ReferenceEquals method, 300
object.System.Object class, 116
objects
heap and, 53
implementing dynamic objects, 860-863
OfType operator, 457
OperationCanceledException, 950
OperationCompleted method, 621
OperationStarted method, 621
operator lifting
equality operators (== and !=), 187
mixing nullable and non-nullable operators, 188
nullable value types, 186
relational operators, 187
operator overloading, 216-219
custom implicit/explicit conversions, 218
operator functions, 216
overloading equality/comparison operators, 217
true/false operators, 219
operator, defined, 26
optional parameters, 61
named arguments and, 991
object initializers versus, 98
order comparison, 249, 306-308
OrderBy operator, 450
OrderByDescending operator, 450
OrderedDictionary, 348
ordering
comparers and collations, 451
IOrderedEnumerable and IOrderedQueryable, 452
LINQ operators and, 450-452
OrderBy and OrderByDescending arguments, 450
PLINQ and, 929
ordinal case-sensitive comparison, 250
ordinal string comparison, 250

OS security, 680-683
- administrative elevation and virtualization, 682
- running in a standard user account, 681

out parameter
- passing, 811
- type names, 801

out parameter modifier, 58
out variables, 14, 58
outer joins
- GroupJoin, 447
- SelectMany, 440

outer variables, 166-169
overflow, 40
overflow check operators
- for constant expressions, 41
- integral types and, 40

overloading
- instance constructors, 94
- operator overloading, 216-219
- overloading methods, 92
- resolution and, 115

override modifier, 111, 113
oversubscription, 591

## P

Parallel class, 940-946
- Parallel.For and Parallel.ForEach, 941-946
- Parallel.Invoke, 940

Parallel Framework (PFX), 923-926
- (see also parallel programming)
- benefits of, 924-926
- components, 925-926
- concepts, 924
- when to use, 926

parallel programming, 923-964
- AggregateException and, 956
- BlockingCollection<T>, 961-964
- concurrent collections, 958-961
- Parallel class, 940-946
- PFX benefits, 924-926
- PLINQ (see PLINQ)
- task parallelism, 946-956

Parallel.For, 941-946
- optimization with local values, 945-946
- ParallelLoopState, 943

Parallel.ForEach, 941-946
- indexed, 942
- optimization with local values, 945-946
- outer versus inner loops, 942
- ParallelLoopState, 943

Parallel.Invoke, 940
ParallelLoopState, 943
parameterless constructor, 115
parameterless constructor constraint, 141
parameters, 53-64
- implications of passing arguments by reference, 59
- in modifier, 60
- named arguments and, 62
- optional parameters, 61
- out modifier, 58
- out variables and discards, 59
- params modifier, 60
- pass-by-value versus pass-by-reference, 93
- passing arguments by value, 56
- ref modifier, 57

params modifier, 60
parentheses (), 26, 65
Parse method, 258, 270
partial methods, 105
partial types, 105
passing by reference
- implications of, 59
- ref modifier, 57

passwords
- hashing, 870
- validation, 1009

Path class, 671-673
pattern variable, 14, 110
patterns, 2, 201-204
- constant pattern, 204
- positional patterns, 203
- property patterns, 201
- tuple patterns, 203
- var pattern, 204

PE (portable executable) assembly, 757
performance counters, 564-568
- creating counters and writing performance data, 568
- enumerating the available counters, 565

reading data from, 566
period (.), 26, 65
PFX (see Parallel Framework)
PipeStream class, 648-652
anonymous pipes, 650-652
named pipes, 648-650
PLINQ (Parallel LINQ), 927-940
canceling a query, 933
custom aggregation optimization, 938-940
functional purity, 932
functionality, 925
input-side optimization, 935-937
limitations, 930
optimizing, 934-940
ordering, 929
output-side optimization, 934
parallel execution ballistics, 929
parallel spellchecker example, 930-932
setting the degree of parallelism, 933
when to use, 932
plug-in methods, writing with delegates, 150
plus sign (+)
addition operator, 39
combining delegate instances, 151
in nested type names, 800
in regular expressions, 1002
string concatenation operator, 47
pointer-to-member operator (->), 220, 221
pointers, 219-223
basics, 220
fixed statement and, 220
fixed-size buffers and, 222
stackalloc keyword and, 221
type names, 801
to unmanaged code, 223
void pointer (void*), 222
polymorphism, 107
POP3 mail, 720-722
portable executable (PE) assembly, 757
ports, TCP/UDP protocols, 690
positional patterns, 203
positive infinity (+∞), 41
positive lookahead, 1003
positive lookbehind, 1004
post-phase action, 911
#pragma warning directive, 225
precedence, operator, 66
predefined types, 28, 35, 55
(see also specific types)
predicate, 374
preempted thread, 577
prefixes
namespaces and, 492
X-DOM, 495
preprocessor directives, 223-225, 1027
Conditional attribute, 225
pragma warning, 225
primary expressions, 65
primitive types, 35
Priority property, 587
private access modifier, 123
private key, 876
private protected access modifier, 123
Process, 311-313, 559, 587
examining running processes, 559
examining threads in a process, 560
Process.Threads property, 560
producer/consumer collection, 959
producer/consumer queue, 962-964
Progress<T>, 627
projecting
LINQ operators, 429-441
Select method, 430-434
SelectMany, 435-441
into an X-DOM, 497-500
properties, 2, 99-101
automatic, 100
calculated properties, 100
CLR property implementation, 101
emitting, 835-837
expression-bodied, 100
get and set accessors, 101
property initializers, 101
read-only, 100
property initializer, 101
property patterns, 201
protected internal access modifier, 123
proxy servers, 701
pseudocustom attributes, 819
public access modifier, 123
public key, 876
public keyword, 30
public-key encryption/signing, 876-879
digital signing, 878
RSA class, 877

punctuators, 26

## Q

quantifiers, 377
  greedy versus lazy, 1003
  LINQ, 466
  regular expressions, 997, 1002
query continuation, 393
query expressions, 2, 378-382, 416-420
  about, 378-380
  building, 416-420
  delegates versus expression trees, 416-420
  expression DOM, 418
  expression trees, 416-420
  mixed-syntax queries, 381
  query syntax versus fluent syntax, 381
  query syntax versus SQL syntax, 381
  range variables, 380
query operators, 369
query strings, 707
querying, X-DOM, 476-481
question mark (?)
  in nullable types, 185
  in regular expressions, 997, 1003
Queue, 340
Queue<T>, 340
quote, single (')
  enclosing char literals, 45
  following generic type names, 800

## R

rainbow tables, 870
Random class, 290
Range operator, 468
range partitioning, 935, 946
range variables, 380
ranges
  arrays and, 50
  using indices and ranges with indexers, 103
RCWs (Runtime-Callable Wrappers), 990
Reactive Extensions, 923
read locks, 898-903
read-only properties, 100
read-only structs, 121
ReaderWriterLockSlim, 898-900
readonly modifier, 90, 121
ReadOnlyCollection<T>, 356
ReadOnlySpan<char>, 970-972
real literals, 37
rectangular arrays, 51
recursive locking, 902
reentrancy, 609
ref locals, 63
ref parameter
  implicit, 992
  passing, 811
  type names, 801
ref parameter modifier, 57
ref returns, 63
ref structs, 122
refactoring, 22
reference assemblies, 235, 1034
reference conversions, 108-110, 146
reference types, 33
referential equality, 296, 298
reflection, 797-849
  anonymously calling members of generic interface, 815
  awkward emission targets, 840-844
  base types and interfaces, 801
  dynamic code generation, 823-830
  emitting assemblies and types, 830-833
  emitting constructors, 837
  emitting fields and properties, 835-837
  emitting generic methods and types, 838-840
  emitting type members, 833-838
  obtaining a type, 798-799
  parsing IL, 844-849
  reflecting and activating types, 798-804
  reflecting and invoking members of a type, 805-817
  reflecting assemblies, 817
  type names, 800
  working with attributes, 818-823
Reflection.Emit object model, 831
Regex...
  Regex object, 999
  Regex.Match, 998
  RegEx.Replace, 1007
  Regex.Split, 1008
  RegexMatchTimeoutException, 998

RegexOptions, 999
RegexOptions.Compiled, 999
RegexOptions.Multiline, 1005
RegisterWaitForSingleObject, 908
regular expressions (Regex), 997-1014
basics, 997-1002
character escapes, 1000
character sets, 1001
compiled regular expressions, 999
cookbook regular expressions, 1009-1012
groups, 1006-1007
language reference, 1012-1014
MatchEvaluator delegate, 1008
named groups, 1007
quantifiers, 1002
RegexOptions, 999
replacing and splitting text, 1007
zero-width assertions, 1003-1006
reimplementing interfaces, 127-129
relational operators, 187
remainder operator (%), 39
Repeat operator, 468
Replace method, 247
reserved keywords, 25
Resize method, 335
ResourceManager class, 771-775
.resources files, 770-772
resources, in assemblies, 768-775
creating a pack URI resource in Visual Studio, 772
directly embedding, 769
.resources files, 770-772
.resx files, 771-772
restore
generic methods, 137
generic types, 136
resurrection, 534-536
.resx files, 771-772
return statement, 80
return types, 22, 92
rich-client applications
application frameworks, 239
threading in, 588-589
right-associative operators, 66
Rijndael class, 871, 875
roots, garbage collection and, 530
Roslyn compiler, 1017-1043
architecture, 1017
compilations and semantic models, 1033-1043
creating a compilation, 1034-1035
emitting an assembly, 1035
querying the semantic model, 1036-1041
scripting, 1018
syntax trees, 1018-1033
workspaces, 1018
Round method, 289
rounding errors, 43
rounding, real to integral conversions, 283
RSA encryption algorithm, 877
runtime assemblies, 1034
runtime type checking, 118
Runtime-Callable Wrappers (RCWs), 531, 990
RuntimeBinderException, 211, 212

## S

satellite assemblies, 773-775
building, 773
cultures and subcultures, 774
testing, 774
Visual Studio designer support, 774
sbyte (numeric type), 41
sealed modifier, 113, 127
searching within strings, 246
security (see cryptography and encryption)
seed factory function, 938
seed value, 938
Select method, 430-434
indexed projection, 431
projecting into concrete types, 433
select subqueries and object hierarchies, 431
subqueries and joins in EF Core, 432-433
selection statements, 72
changing the flow of execution with braces, 73
else clause and, 73
if statement, 72
switch expressions, 77
switch statements, 74-75
switching on types, 75
SelectMany, 435-441

- in EF Core, 439-440
- joining with, 438
- multiple range variables, 436
- outer joins with, 440
- thinking in query syntax, 437

semantic model
- declared symbols, 1039
- looking up symbols, 1040
- symbol accessibility, 1038
- SymbolInfo, 1038
- symbols, 1037-1039
- for syntax tree, 1036-1041
- TypeInfo, 1039

semaphore, 896-898
- asynchronous semaphores and locks, 897
- writing an EnterAsync extension method, 898

semicolon (;), 22, 26
SendAsync method, 698
SequenceEqual method, 467
sequences, 369
serialization, 725-754
- attribute-based, 729-731
- binary serializer, 747-749
- concepts, 725-729
- defined, 204
- engine comparison, 726-728
- explicit versus implicit, 729
- formatters, 728
- JSON serializer, 738-747
- XML serializer, 729-738

serialization engines, 726-728
- binary serializer, 728
- data contract serializer, 728
- IXmlSerializable hook, 728
- JsonSerializer, 727
- XmlSerializer, 727

set accessor, 101
set operators (LINQ), 456
- Concat and Union, 456
- Intersect and Except, 457

SetData method, 916
SetValue method, 330, 485
shared memory, 982-984
- (see also memory-mapped files)

shared state, 576
shared writable state, 582
shift left operator (<<), 41
shift right operator (>>), 41
SignalAndWait method, 909
signaling
- event wait handles for, 903-910
- threading, 587

signature, 92
signing, digital, 878
signtool.exe, 767
single dispatch, 857
Single operator, 461
single-line comments, 27
single-threaded program, 576
single-threaded timers, 920
Skip operator, 428
SkipWhile operator, 429
slicing
- defined, 965
- spans and, 966-969

SmtpClient, 716
SortedDictionary<TKey,TValue>, 349
SortedSet<T>, 342
Span<T> struct, 965-969
spans
- CopyTo and TryCopyTo, 967
- forward-only enumerators, 970-972
- slicing and, 966-969
- stack-allocated/unmanaged memory, 972
- working with text, 968

spinning, blocking versus, 579
Split method, 248
splitting strings, 248
square brackets ([])
- array declaration, 23, 48, 51
- in regular expressions, 1000

stack, 53
Stack, 341
stack-allocated memory, 972
Stack<T>, 341
stackalloc keyword, 221
StackFrame class, 560-562
StackTrace class, 560-562
StartsWith method, 246
state, 864
statement block, 22, 70, 166
statements, 21, 70-81
- declaration statements, 71
- expression statements, 71
- iteration statements, 77-79

jump statements, 79-81
selection statements, 72
switch statements, 74-75
static binding, 208
static classes, 104, 193-195
static constructors, 103
static local methods, 93
static members
instance versus, 29
locking, 892
static methods, 193-195
static readonly field, 91
static type checking, 118
static types, 214
static typing, 2
static variable flags, 552
Stop method, 525
Stopwatch class, 569
StorageFile class, 676-677
StorageFolder class, 676-677
stream adapters, 653-661
binary adapters, 659
closing and disposing, 660
text adapters, 654-658
Stream...
Stream class, 639
StreamReader, 656, 719
StreamSocket class, 722
StreamSocketListener class, 722
StreamWriter, 656, 719
streams and I/O, 637-686
backing store streams, 643
BufferedStream, 652
closing and flushing, 642
compression streams, 661-664
file and directory operations, 665-675
file I/O in UWP, 676-680
FileStream, 644-647
memory-mapped files, 683-686
MemoryStream, 647
OS security, 680-683
PipeStream class, 648-652
reading and writing, 641
seeking, 642
stream adapters, 653-661
stream architecture, 637-639
thread safety, 643
timeouts, 643
using streams, 639-653
ZIP files, 664
string type, 27, 46-48, 245-256
accessing characters within, 246
comparing, 249-252
constructing strings, 245
enum conversions, 294
manipulating, 247
null/empty, 246
ordinal versus culture comparison, 250
searching within strings, 246
spans and, 968
splitting/joining, 248
string equality comparisons, 250
string order comparison, 251
String.format and composite format strings, 248
StringBuilder class, 252
text encodings and Unicode, 253-256
String...
string.Format, 248, 273
string.IsNullOrEmpty, 246
StringBuilder class, 236, 252
StringComparer, 365
StringComparison enum, 251
StringInfo class, 256
StringReader, 658
StringSplitOptions enum, 248
StringWriter, 658
strongly named assemblies, 762
strongly typed language, 2
struct constraint, 141
structs, 120-122
construction semantics, 120
mapping to unmanaged memory, 984-988
mapping to unmanaged method, 978-980
read-only structs/functions, 121
ref structs, 122
speeding up equality comparison with, 302
structural comparison, 366
structural equality, 296
structured parallelism, 924
subclass
reimplementing an interface in, 127-129
subclassed collection elements, 736

subclasses and child objects, 731-734
subclassing child objects, 733
subclassing generic types, 141
subclassing serializable classes, 753
subclassing the root type, 731
subcultures, 774
subqueries, 388-391
deferred execution and, 391
select subqueries and object hierarchies, 431
subqueries and joins in EF Core, 432-433
subscribers, 158
Substring method, 247
Sum operator, 463
surrogate pairs, 255
switch expressions, 77
switch statements, 74-75, 204
SymbolInfo, 1038
symbols
renaming a symbol, 1041-1043
syntax trees and, 1037-1038
symmetric encryption, 871-876
chaining encryption streams, 873
disposing encryption objects, 875
encrypting in memory, 872
key management, 875
synchronization context scheduler, 955
synchronization object, 885
SynchronizationContext class, 589
synchronous call graph, 602
synchronous completion, 621-623
syntax trees, Roslyn, 1018-1033
common properties and methods, 1021
CSharpSyntaxRewriter, 1032
CSharpSyntaxWalker, 1026
declared symbols, 1039
finding a child by its offset, 1025
handling changes to source code, 1030
looking up symbols, 1040
node types, 1020
obtaining, 1022
preprocessor directives, 1027
renaming a symbol, 1041-1043
structure, 1019-1021
structured trivia, 1028
symbol accessibility, 1038
SymbolInfo, 1038
symbols, 1037-1038
SyntaxFactory and, 1030-1032
transforming, 1029-1033
traversing children, 1023-1025
traversing parents, 1025
traversing/searching a tree, 1023-1026
trivia, 1027-1029
TypeInfo, 1039
working with TextSpan, 1025
syntax, C#, 24-27
comments, 27
identifiers and keywords, 25-26
literals, punctuators, and operators, 26
Syntax...
SyntaxFactory, 1030-1032
SyntaxFactory.Token, 1031
SyntaxNode, 1019, 1023-1025, 1029
SyntaxToken, 1019
SyntaxTree, 1025
SyntaxTrivia, 1019, 1025, 1028
System...
System.AppContext, 313
System.ArgumentException, 177
System.ArgumentNullException, 178
System.ArgumentOutOfRangeException, 178
System.Attribute, 205, 819, 820
System.Buffers, 541
System.Buffers.MemoryPool<T>, 970
System.Buffers.Text, 969
System.Char, 243
System.Collections.*, 749
System.Collections.Concurrent, 535, 958
System.Collections.Generic, 180
System.Collections.Generic.IEnumerable<T>, 179
System.Collections.Generic.IEnumerator<T>, 179
System.Collections.IEnumerable, 179, 180
System.Collections.IEnumerator, 179
System.Collections.Immutable, 357
System.ComponentModel, 589, 634
System.Core.dll, 852
System.Data, 526
System.Data.IDataRecord, 862
System.Data.SqlTypes.SqlBoolean, 219
System.Delegate, 152

System.Diagnostics, 311-313, 558, 587
System.Diagnostics.PerformanceCounter, 564
System.Drawing, 526
System.Dynamic, 851
System.Enum, 291
System.Environment class, 310
System.EventArgs, 160
System.EventHandler<>, 161
System.Exception, 172, 177
System.GC.Collect, 536
System.Globalization, 256, 260
System.Globalization.CultureInfo, 775
System.IDisposable, 174
System.InvalidOperationException, 178
System.IO, 526, 665, 687
System.IO.Compression, 661, 664
System.Linq.Enumerable class, 812
System.Linq.Expressions, 1018
System.Management, 667
System.MulticastDelegate, 152
System.Net, 609, 687, 690
System.Net.Mail, 716
System.NotImplementedException, 178
System.NotSupportedException, 178
System.Nullable<T>, 185
System.Object, 116, 303
System.ObjectDisposedException, 178
System.Reflection, 761
System.Reflection.Emit, 797, 823, 831
System.Runtime, 570
System.Runtime.CompilerServices, 851
System.Runtime.InteropServices, 994
System.Runtime.Loader, 776
System.Runtime.Serialization.Formatters.Binary, 748
System.Security.AccessControl, 682
System.Security.Cryptography, 291
System.Security.Cryptography.X509Certificates, 868, 879
System.Security.Cryptography.Xml, 868
System.String, 245, 307
System.Text, 236, 254, 1030
System.Text.Encodings.Web. JavaScriptEncoder, 747
System.Text.Json, 738
System.Text.Json.JsonDocument, 519
System.Text.Json.Serialization, 741
System.Text.Json.Utf8JsonReader, 516-518
System.Text.Json.Utf8JsonWriter, 518
System.Text.RegularExpressions, 236, 997
System.Threading, 544
System.Threading.Channels.Channel, 923
System.Threading.Tasks, 593
System.Threading.Timer, 919
System.Timers, 543, 919
System.Timers.Timer, 540
System.Tuple, 200
System.Type, 118, 798
System.Uri class, 202
System.WeakReference class, 545
System.Windows.Forms.Timer, 920
System.Windows.Threading.DispatcherTimer, 920
System.Xml.Linq, 483, 750
System.Xml.Serialization, 205, 729

## T

Take operator, 428
TakeWhile operator, 429
TAP (Task-Based Asynchronous Pattern), 629
task combinators, 629-633
    custom combinators, 631
    WhenAll, 630
    WhenAny, 629
task parallelism, 946-956
    canceling tasks, 949
    child tasks, 948
    continuations, 950-954
    creating and starting tasks, 947
    defined, 924
    specifying a state object, 947
    task schedulers, 955
    TaskCreationOptions, 948
    TaskFactory, 955
    waiting on multiple tasks, 949
task schedulers, 955
Task-Based Asynchronous Pattern (TAP), 629

Task...
  Task class, 592
    (see also tasks)
  Task.Delay, 600
  Task.Factory, 955
  Task.Factory.StartNew, 599, 947
  Task.Run, 593, 602, 947
  Task.WaitAll, 949
  Task.WaitAny, 949
  Task.WhenAll, 630
  Task.WhenAny, 629
  Task<T>, 617
  Task<TResult>, 594, 612, 951
  TaskCanceledException, 950
  TaskCompletionSource, 598-600, 604, 612, 614
  TaskCreationOptions, 948
  TaskCreationOptions.LongRunning, 594, 599
  TaskFactory object, 955
  TaskScheduler.UnobservedTaskException, 596
tasks, 592-600
  asynchronous programming and, 602
  continuations, 596-597
  exceptions, 595
  exceptions and autonomous tasks, 595
  long-running, 594
  returning values, 594
  starting a task, 593-594
  TaskCompletionSource, 598-600
  Wait method, 593
TCP (Transmission and Control Protcol)
  basics, 717-720
  concurrency with, 719
  receiving POP3 mail with, 720-722
  in UWP, 722-723
TcpClient, 717-720
TcpListener, 717-720
text
  MatchEvaluator delegate, 1008
  replacing and splitting with regular expressions, 1007
  spans and, 968
text adapters, 654-658
  character encodings, 657-658
  StreamReader and StreamWriter, 656
  StringReader and StringWriter, 658
text encoding, 253-256
  encoding to byte arrays, 255
  file and stream I/O, 255
  obtaining an Encoding object, 254
  UTF-16 and surrogate pairs, 255
text handling, 243-245
  char type, 243
  text encodings and Unicode, 253-256
TextSpan, 1025
ThenBy operator, 451
ThenByDescending operator, 451
thin-client applications, 239
this keyword, 94
this reference, 98
thread execution barrier, 910
thread pool, 590
  entering, 591
  hygiene in, 591
thread safety, 643
thread-local storage, 914
  AsyncLocal<T>, 917
  GetData and SetData, 916
  ThreadLocal<T>, 915
  ThreadStatic attribute, 915
thread-safe code, 582
thread-safe objects, 892
thread-unsafe operations, 932
Thread...
  Thread object, 576
  Thread.Sleep, 578
  ThreadLocal<T>, 915, 931
  ThreadPool.RegisterWaitForSingleObject, 908
  ThreadStart delegate, 576
  ThreadStatic attribute, 915
threading, 576-592, 881-921
  advanced topics, 881-921
  Barrier class, 910-911
  blocking, 578
  blocking versus spinning, 579
  creating a thread, 576-578
  exception handling, 584
  exclusive locking, 882-890
  foreground versus background threads, 586
  I/O bound versus compute-bound operations, 579
  join and sleep, 578
  lambda expressions and captured variables, 584

lazy initialization, 912-914
limitations of, 592
local versus shared state, 580-582
locking and thread safety, 582, 890-895
multiple UI threads, 589
nonexclusive locking, 896-903
passing data to a thread, 583-584
Priority property, 587
in rich-client applications, 588-589
signaling, 587
signaling with event wait handles, 903-910
synchronization contexts, 589
synchronization overview, 882
thread pool, 590
thread-local storage, 914
timers, 918-921
threads
defined, 576
examining threads in a process, 560
throw expressions, 16, 176
throw statement, 81
throwing exceptions, 175-177
rethrowing exceptions, 176
throw expressions, 176
tiered compilation, 232
time zones, 264-269
DateTime and, 264
DateTimeOffset and, 265
Daylight Saving Time and DateTime, 268
TimeZone and TimeZoneInfo, 265-268
time-stamping server, 768
timeouts, 643
timers, 918-921
memory leaks and, 543
multithreaded, 919-920
single-threaded, 920
TimeSpan, 256
TimeZone, 265-268
TimeZone.CurrentTimeZone method, 265
TimeZoneInfo, 265-268
To...
ToArray operator, 459
ToDictionary operator, 459
ToHashSet operator, 459
ToList operator, 459
ToLocalTime, 264
ToLongDateString method, 263
ToLookup operator, 459
ToLower, 244, 248
ToShortDateString method, 263
ToString method, 119, 252, 263, 270, 473
ToUniversalTime, 264, 269
ToUpper, 244, 248
Trace class (see Debug and Trace classes)
TraceFilter, 557
TraceListener, 556
traces, 571
transport layer, 687
trivia
preprocessor directives, 1027
structured, 1028
syntax trees and, 1027-1029
unstructured, 1028
try statements and exceptions, 170-179
alternatives to exceptions, 179
catch clause, 172-173
common exception types, 177
finally block, 174
key properties of System.Exception, 177
throwing exceptions, 175-177
try/catch/finally blocks, 183
TryXXX method pattern, 178
using declarations, 175
using statement, 174
TryCopyTo method, 967
TryEnter method, 884
TryParse, 258, 270
TryXXX method pattern, 178
tuple literal, 197
tuple patterns, 203
tuple types, 197
TupleElementNamesAttribute, 199
tuples, 197-200
deconstructing, 199
equality comparison, 200
in C# 7, 16
naming tuple elements, 198
System.Tuple classes, 200
type erasure, 198
ValueTuple.Create, 199
type arguments, 136

type checking, 118
type converters, 285
type equivalence, 994
type library importer, 990
type marshaling, 976-980
    in and out marshaling, 980
    marshaling classes and structs, 978-980
    marshaling common types, 976-978
type members, emitting, 833-838
type parameters
    covariance, 145
    declaring, 138
    generic types and, 136
type patterns, 14
type safety, 2, 133
type system, C#
    access modifiers, 123-124
    anonymous types, 195
    basics, 27-35
    Boolean type and operators, 43-45
    C# members versus CLR members, 808
    converting types, 30
    creating types, 89-148
    custom type examples, 28-30
    dynamically invoking a member, 810
    emitting assemblies and types, 830-833
    enums, 131-134
    extension methods, 193-195
    generics, 135-148
    inheritance (see inheritance)
    instantiating a type, 802-803
    interfaces, 125-130
    nested types, 134-135
    numeric types, 36-43
    object type, 116-120
    predefined type examples, 27
    strings and characters, 45-48
    structs, 120-122
    value types versus reference types, 31-35
type unification, 116, 292
Type...
    TypeAttributes, 831
    TypeBuilder, 831, 832, 835
    TypeIdentifierAttribute, 994
    TypeInfo class, 799, 805, 1039
typeof operator, 118, 139
types
    aliasing within namespaces, 85
    base types and interfaces, 801
    partial types/methods, 105
    reflecting and activating, 798-804
    reflecting and invoking members of, 805-817

## U

UAC (User Account Control), 680
UI (see user interface)
unbound generic type, 139
unboxing, 117
    is operator and, 110
    nullable values, 186
#undef directive, 552
Unicode, 253-256
    UTF-16 and surrogate pairs, 255
    XmlWriter and, 490
UnicodeCategory enum, 245
unified type system, 1
union, 981
Union operator, 456
Universal Windows Platform (UWP) (see UWP)
Unix gzip file compression, 663
Unix, OS security, 680
unmanaged code
    callbacks from, 980
    pointers to, 223
unmanaged constraint, 141
unmanaged heap, 985
unmanaged memory, 972, 984-988
UnmanagedType, 976
unnamed methods, 616
/unsafe compiler option, 219
unsafe code, 219-223
unseeded aggregations, 464
upcasting, 108
upgradeable locks, 901-903
UploadValues method, 707
Uri class, 690-692
URIs, 690-692
User Account Control (UAC), 680
user interface (UI)
    awaiting in, 608-610
    multiple UI threads, 589

UseShellExecute, 311
ushort (numeric type), 35
using declarations, 175
using directive, 82, 85
using statement, 81
using static directive, 83
UTC (Coordinated Universal Time), 258, 265
Utf8JsonReader, 516-518
Utf8JsonWriter, 518
utility classes, 309-313
  AppContext, 313
  Console, 309
  Environment, 310
  Process, 311-313
UWP (Universal Windows Platform), 241
  about, 5
  application folder, 678
  downloads folder, 679
  file I/O in, 676-680
  isolated storage, 678
  KnownFolders class, 678
  obtaining directories and files, 677
  removable devices, 679
  TCP in, 722-723
  user-selected files and folders, 679
  working with directories, 676
  working with files, 677

## V

value equality, 296
value types, 32
ValueTask<T>, 617, 623
ValueTuple.Create, 199
ValueTuple<string,int>, 199-200
var keyword, 64
var pattern, 204
var type, dynamic type versus, 212
variables, 53-64
  (see also parameters)
  default values, 55
  definite assignment and, 54
  heap, 53
  purpose of, 27
  ref locals, 63
  ref returns, 63
  stack and, 53
  var keyword, 64
verbatim string literals, 47
vertical bar (|)
  bitwise OR operator, 41, 132
  in regular expressions, 998
view accessors, 685
virtual function members, 111
virtualization, 682
Visitor pattern, 854-857
void expressions, 65
void pointer (void*), 222
volume information, querying, 674

## W

wait handles (see event wait handles)
Wait method, 593
WaitAll method, 909
WaitAny method, 909
#warning preprocessor directive, 224
WCF (Windows Communication Foundation), 727
weak references
  caching and, 546
  events and, 547-549
  GC and, 545-549
WebClient, 693-694
  custom headers, 706
  uploading form data, 707
WebException, 704
WebExceptionStatus enum, 704
WebRequest, 695-696
  custom headers, 706
  uploading form data, 708
WebRequest.RegisterPrefix, 696
WebRequestMethods.Ftp, 713
WebResponse, 695-696
Where clause, 426-428
  Enumerable.Where implementation, 426
  indexed filtering, 427
WHERE clause, 428
while loops, 77
wildcards (character sets), 1001
Windows
  application manifest, 759
  memory-mapped files and shared memory, 684
  OS security, 680

Windows Communication Foundation (WCF), 727
Windows Data Protection API (DPAPI), 868
Windows Desktop application framework, 240
Windows event logs, 562-564
 monitoring, 564
 reading, 563
 writing to, 562
Windows Forms, 241
Windows Management Instrumentation (WMI) API, 667
Windows Runtime (WinRT)
 asynchronous methods in, 619
 event semantics in, 160
 garbage collection and, 531
 interoperability with C#, 7
Windows.Networking.Sockets, 722
WithDegreeOfParallelism, 933
WithMergeOptions, 929
WMI (Windows Management Instrumentation) API, 667
word boundary assertions, 1005
WPF, 240
write locks, 898-903

## X

x++ (incrementing), 582
Xamarin, 5, 242
XAML (Extensible Application Markup Language), 285
XAML files, 772
XAttribute, 474
XContainer, 472
XDeclaration object, 487
XDocument, 472, 487-489
XElement, 470-474
 using XmlReader with, 514
 using XmlWriter with, 515
XML declarations, 489
XML documentation, 226-229
 standard tags, 226-228
 type or member cross-references, 229
 user-defined tags, 228
XML DOM (X-DOM), 470
 attribute navigation, 481
 automatic deep cloning, 476
 automatic XText concatenation, 487
 child node navigation, 477
 content specification, 475
 default namespaces, 494
 functional construction, 474
 getting values, 485
 instantiating, 474-476
 loading and parsing, 472
 mixing XmlReader/XmlWriter with, 514
 namespace specification, 493
 navigating and querying, 476-481
 overview, 470-474
 parent navigation, 480
 peer node navigation, 481
 prefixes, 495
 projecting into, 497-500
 removing a sequence of nodes or attributes, 483
 retrieving a single element, 479
 retrieving descendants, 479
 retrieving elements, 478
 saving and serializing, 473
 setting values, 485
 simple value updates, 482
 updating, 481-484
 updating child nodes and attributes, 482
 updating through the parent, 483
 values and mixed content nodes, 486
 working with values, 484-487
XML serializer, 729-738
XmlConvert, 284
XmlReader, 501-509
 mixing with an X-DOM, 514
 namespaces and prefixes, 508
 patterns for using, 511-516
 reading attributes, 507
 reading elements, 503-507
 reading nodes, 502
 using with XElement, 514
 working with hierarchical data, 511-514
XmlSerializer, 729-738
 attribute-based serialization, 729-731
 attributes, names, and namespaces, 730
 basics, 727
 IXmlSerializable, 736-738

- serializing child objects, 732
- serializing collections, 734-736
- subclassed collection elements, 736
- subclasses and child objects, 731-734
- subclassing child objects, 733
- subclassing the root type, 731
- XML element order, 731

XmlWriter, 509
- mixing with an X-DOM, 514
- patterns for using, 511-516
- using with XElement, 515
- working with hierarchical data, 511-514
- writing a declaration to a string, 490

XNode, 472

XObject, 470, 496

XStreamingElement, 500

XText, 487

## Y

yield break statement, 182

## Z

zero-width assertions, 1003-1006
- anchors, 1004
- defined, 1003
- lookahead and lookbehind, 1003
- word boundaries, 1005

ZIP files, 664

Zip operator, 449

ZipArchive class, 664

ZipFile class, 664

# About the Authors

**Joseph Albahari** is the author of *C# 7.0 in a Nutshell*, *C# 7.0 Pocket Reference* and *LINQ Pocket Reference*. He also wrote LINQPad—the popular code scratchpad and LINQ querying utility.

**Eric Johannsen** is a software development and machine learning leader equally at home with business and technology. He's experienced in architecting software solutions to fulfill emerging business needs as well as identifying new business opportunities enabled by technology.

# Colophon

The animal on the cover of *C# 8.0 in a Nutshell* is a numidian crane. The numidian crane (*Antropoides virgo*) is also called the demoiselle crane because of its grace and symmetry. This species of crane is native to Europe and Asia and migrates to India, Pakistan, and northeast Africa in the winter.

Though numidian cranes are the smallest cranes, they defend their territories as aggressively as other crane species, using their loud voices to warn others of trespassing. If necessary, they will fight. Numidian cranes nest in uplands rather than wetlands and will even live in the desert if there is water within 200 to 500 meters. They sometimes make nests out of pebbles in which to lay their eggs, though more often they will lay eggs directly on the ground, protected only by vegetation.

Numidian cranes are considered a symbol of good luck in some countries and are sometimes even protected by law. Many of the animals on O'Reilly covers are endangered; all of them are important to the world.

The cover illustration is by Karen Montgomery, based on a black and white engraving from *Wood's Illustrated Natural History*. The cover fonts are Gilroy Semibold and Guardian Sans. The text font is Adobe Minion Pro; the heading font is Adobe Myriad Condensed; and the code font is Dalton Maag's Ubuntu Mono.

O'REILLY®

There's much more where this came from.

Experience books, videos, live online training courses, and more from O'Reilly and our 200+ partners—all in one place.

Learn more at oreilly.com/online-learning

©2019 O'Reilly Media, Inc. O'Reilly is a registered trademark of O'Reilly Media, Inc. | 175