### Symbols

- `{ } (curly brackets), 34
- `^` = assignment operator (Boolean), 62
- `=` = assignment operator (Boolean), 60
- `!` = comparison operator (Boolean), 60
- `||` = conditional operator (Boolean), 61
- `:` = operator, namespaces and, 360–361
- `< > angle brackets, syntax, 320
- `%=` assignment operator, 5
- `*=` assignment operator, 52
- `+=` assignment operator, 52
- `-=` assignment operator, 52
- `/=` assignment operator, 52
- `&=` assignment operator (Boolean), 62
- `<` = comparison operator (Boolean), 60
- `<=` = comparison operator (Boolean), 60
- `>` = comparison operator (Boolean), 60
- `>=` = comparison operator (Boolean), 60
- `&&` = conditional operator (Boolean), 60
- `%` operator, 47
- `*` operator, 47
- `+` operator, 47
- `-` operator, 47
- `/` operator, 47
- `??` (null coalescing) operator, 323–324
- `?.` (null condition) operator, 325
- `/* with comments, 35
- `// with comments, 36
- `2D rendering, XAML and, 426
- `3D rendering, XAML and, 426

### A

- `About window (KarliCards), 453–458
- `absolute path names, 646–647
- `abstract classes, 204, 226–228
- `abstract keyword, 204, 250–251
- `AcceptsReturn property, 459
- `accessibility, 182
- `accumulators, 418
- `Add() method, 283
- `Add New Item Wizard, 219–220
- `AddRange() method, 276
- `ADO.NET (Active Data Objects .NET), 11
- `Entity Framework and, 732
- `Aggregate() method, 416, 418
- `aggregate operators, LINQ, 717–720
- `aliases, namespaces, 360–361
- `alignment, 442
- `AllowMultiple property, 385–386
- `animation, WPF, 484–485
- `anonymous methods, 375, 409–413
- `anonymous types, 394–398
- `AOT (Ahead of Time) compiler, 580
- `NGEN and, 580
- `API Portability Analyzer, 605
- `APIs (Application Programming Interfaces), 560
- `applications, 10
- `cross-platform programs and, 581
- `Append() method, 335
- `AppendChild() method, 685
- `AppendFormat() method, 335
- `ApplicationException class, 361
- `applications
- `APIs, 10
- `ASP.NET, web applications, 548–555
- `Cloud/Web, 10
- `console applications, 16, 21–23, 32
- `desktop, 10, 16, 26–31, 32, 196–199
- `event-driven, 195
- `.NET Core, 5–9, 13
- `.NET Framework, 5–9, 13
- `UWP (Windows Universal Platform), 4
- `badges, 796–797
- `creating, 794–795
applications (continued)
disk access, 807–810
lifetime, 797
Lock Screen, 797
menus, 796
navigation, 810–812
sandboxed, 805–807
screen orientation, 796
tiles, 796–797
toolbars, 796
WCF services, 11
Web APIs, 10
Windows Store, 10, 816–819
WPF (Windows Presentation Foundation), 10
writing, 5–6
archived files, 661
ArrayList collection, 274, 320
arrays, 84, 99, 100–101, 103–106
of arrays, 108–109
collections, 194
declaring, 99–100
expression filter, 103
foreach loops, 101, 105
int, 124
multidimensional, 106–108
parameters, 124–126
pattern matching, switch statement and, 102–103
as operator, 315–317
ASP.NET, 533
code-behinds, 611
controls, HTML and, 609
selecting, 611–612
web applications, 548–555
ASP.NET Core, 612, 617–618
authentication, 631–632
authorization, 631–632
self-hosting and, 628
web applications, 627–637
ASP.NET MVC, 612, 614–616
ASP.NET Web API, 617
ASP.NET MVC (Model, View, Controller) and, 560
consuming from web site, 567–571
creating, 560–564
deployment to cloud, 564–567
JSON file, Newtonsoft.Json library and, 568–570
scaling, 572–575
ASP.NET Web Development Server, 548–549
ASP.NET Web Forms, 611, 613–614
authentication, 623–627
authorization, 623–627
input, validation, 621–622
server controls, 620–621
state management, 622
ASP.NET Web Site, ASP.NET Web Application Project
types comparison, 618–620
assemblies, 7, 581
assignment operators, 52–53
%\=, 52
\*=, 52
+=, 52
-=, 52
/=, 52
=, 42, 52
Boolean, 62
overloading, 301
async keyword, 168
asynchronous file access, 660–661
attached property, 435–436
attributes, 384
AttributeUsageAttribute, 385–386
creating, 385–386
OptionalAttribute, 404
reading, reflection and, 384–385
using statement, 384
AttributeUsageAttribute attribute, 385–386
automatic properties, 243–244
Average() method, 717
await keyword, 168
Azure, 538
platform, 540
SQL Azure, 538
storage container
ASP.NET web site, 548–555
catch block, 543–544, 546
creating, 539–540
try block, 543–544, 546
Azure Active Directory, 538
Azure Cloud Services, 538
Azure Storage, libraries, 540–547
Azure VM, 538
B
base class methods, 244–247
base keyword, 214, 246
BCL (Base Class Libraries), 5
cross-platform programs and, 582
.NET Standard and, 583
binary operators, overloading, 301
blocks, 34
  indentation, 34
bool type, 40, 60
Boolean logic, 59–64
branching, 59
  if statement, 65–69
  switch statement, 69–72
ternary operators, 65
break command, 79
bubbling events, 438
Button control, 430, 441
byte integer type, 39

C
C# overview, 9–11
Call Hierarchy view, 218
Card class, 255
CardLib. See also KarlLibs
  Cards collection, 279–282
class libraries, 256–265
custom exceptions, 362–363
Deck.GetCard( ) method, 362–363
deep copying, 290–292
event handlers, 375–383
generic collection classes, 339
  operator overloads, 302–307
case sensitivity, 36
catch blocks, 168–175
char type, 40
char variables, implicit conversion, 48
CheckBox control
  Content property, 460
  IsChecked property, 460
  IsThreeState property, 460
checked keyword, 314
CheckForMessage( ) method, 370
CIL (Common Intermediate Language), 6, 8
  CLR and, 580
Circuit Breaker cloud programming pattern, 538
class diagrams, 220–222
class keyword, 204–206
class libraries, 222–226
  CardLib, 256–265
Class View, 216–218
classes
  abstract, 204
  access modifiers, 206
  adding, 219–220
ApplicationException, 361
base classes, 244–247
Card, 255
ConnectionBase, 276
collections, 193, 194, 270
ComputerPlayer, 504–511
Connection, 368–369
containment, 193–194
Deck, 255–256
defining, 204–209
  partial definitions, 252–253
DeflateStream, 642
dependency properties, implementing, 486–499
exception classes, 361
file system access, 642
FileStream, 648–655
FileSystemWatcher, 664–669
generic, 320
GZipStream, 642
instances, 181
interface implementation, 250–252
internal, 204
members, 234
MessageArrivedEventArgs, 372
Panel, 444
Player, 504–511
sealed, 204
static, 185–186
StreamReader, 648, 658–660
StreamWriter, 648, 655–657
versus structs, 228–230
System.Attribute, 386–387
System.Timers.Timer, 366
UML and, 181
Vector, 328–329
XmlAttribute, 678
XmlComment, 678
XmlDocument, 677, 678
XmlNodeList, 678
XmlText, 678
Clear( ) method, 276
Clone( ) method, 289
Cloner object, 290
cloud
  applications, 10
  hybrid cloud, 534–535
IaaS (Infrastructure as a Service) and, 536
cloud (continued)
.NET Framework and, 536
PaaS (Platform as a Service) and, 536
private cloud, 534–535
public cloud, 534–535
SaaS (Software as a Service) and, 536
service models, 535
cloud optimized stack, 536, 560, 591
cloud programming
dependencies and, 533–534
elasticity, 534, 536
patterns
  Circuit Breaker, 538
  CQRS, 537
  Health Endpoint Monitoring, 538
  Materialized View, 537
  Retry, 538
  Sharding, 537
  Throttling, 538
  Valet Key, 537
portability, 536, 537
resiliency, 536, 538
scalability, 536, 537
small footprint, 536
CLR (Common Language Runtime), 5, 7
  CIL conversion and, 580
  cross-platform programs and, 582
  virtual machine, 580
code
  accessibility, 182
  compiling, 6, 8–9
  decorating, 384
  managed code, 7
  native code, 6
  reusable, 118
Code First
database
  creating, 733–740
  location, 740–742
  relationships, 742–749
  Entity Framework and, 732
code-behind files, 427, 429
ASP.NET, 611
collection classes, 270
  strongly typed, 270
collection initializers, 389–390
CollectionBase class, 276
collections, 193, 194, 269–270
  ArrayList, 274
  arrays, 194, 271–276
defining, 276–277, 317
implementing, 278–279
iterators and, 288
key values, 282–283
lambda expressions and, 416–419
sorting, 309–310
  Ch11Ex05, 310–313
column layout, 451–452
COM (Component Object Model), 398
ComboBox control, 461–462
comments, 35
  /*, 35
  //, 36
Compare( ) method, 308, 335
CompareTo( ) method, 308
comparison operators
  Boolean, 60
  overloading, 301
comparisons, 270
type
  boxing, 292–294
  is operator, 294–296
  unboxing, 292–294
value, operator overloading and, 298–302
compiling, 6, 8–9
  JIT (just-in-time), 6
compressed files, reading/writing, 661–664
ComputerPlayer class, 504–511
conditional operators, 4
  Boolean logic, 61
Connect( ) method, 369–370
Connection class, 368–369
Console App (.NET Core), 602–605
course applications, 16, 32
creation, 21–23
ConsoleApplication1, 23, 36–37
  Solution Explorer, 24
  syntax, 36–38
const pattern, 297
constructors, 211–212
  AttributeTargets parameter, 385–386
  execution sequence, 212–216
  FrameworkPropertyMetadata, 487–488
  initializer, 214
containment, 193–194
Content property, 441, 460, 461
contravariance, 353, 354–355
control events, 437–438
controls (WPF)
Border, 443
Button, 430
Canvas, 430
Grid, 441
Label, 430
layout, 441–442
StackPanel, 430
TextBox, 430
types, 441
conversions, 270, 313–316
Copy( ) method, 643
copying
  deep copies, 289, 230–231
  objects, shallow versus deep, 230–231
CoreCLR, cross-platform programs and, 582
CoreFX, cross-platform programs and, 583
CoreRT, cross-platform programs and, 583
Count( ) method, 717
covariance, 353, 354
CQRS (Command and Query Responsibility Segregation) cloud programming pattern, 537
Create( ) method, 643
CreateAttribute( ) method, 685
CreateComment( ) method, 685
CreateDirectory( ) method, 643
CreateElement( ) method, 685
CreateNode( ) method, 685
CreateTextNode( ) method, 685
cross platform, .NET Core and, 590–591
cross-platform programs
  APIs, 581
  assemblies, 581
  BCL (Base Class Libraries), 582
  CLR (Common Language Runtime), 582
  CoreCLR, 582
  CoreFX, 583
  CoreRT, 583
cross platform, 581
dependencies, 582
ecosystems, 581
forking, 582
frameworks, 581
GitHub, 582
hardware platform and, 581
libraries, 581
metadata, 582
metapackages, 582
.NET Native, 583
NuGet, 582
open source, 581
packages, 582
PCL (Portable Class Libraries), 582
runtime, 582
semantic versioning, 582
software platform and, 581
stacks, 581
standards, 581
target framework, 582
TFM (Target Framework Moniker), 582
versions, 582
verticals, 581
CTS (Common Type System), 5
custom exceptions, 361
CardLib project, 362–363
data binding, Options window (KarliCards), 467–472
data languages, 673
JSON, 674–675
XML, 674
data templates, 497
databases, 731–732
  Code First, 733–742
  entity-relationship model, 732
  migration, 749–750
  relational databases, 731
  relationships, 742–749
  SQL Azure, 538
  XML, 750–756
DataContext control, 468
debugging, 22–23, 150
  Break mode
    assertions, 161–162
    breakpoints, 159–160
    entering Break mode, 158–159
    stepping through code, 164–166
    variable content monitoring, 162–164
  Call Stack window, 167
  Command window, 166
  Immediate window, 166
Normal mode
diagnostics output, 158
Output window, 151–156
tracepoints, 156–158
WPF apps, 443–444
Deck class, 255–256
Deconstruct( ) method, 242
decorating code, 384
decrement operators, 48
deep copies, 230–231
CardLib, 290–292
MemberwiseClone( ) and, 289
default keyword, generic class definition, 342
defining classes, partial, 252–253
defining collections, 276–277
defining methods, partial, 253–255
DeflateStream class, 642
delegate keyword, 410
delegate types, 374
delegates
  event handlers, 363
  functions, 144–147
  lambda expressions as, 415–416
Delete( ) method, 643
dependencies, cross-platform programs and, 582
dependency property, 431, 435
  implementing, 486–499
Register( ) method, 486
Design View, WYSIWYG view, 429
desktop applications, 10, 16, 26–32, 196–199
destructors, 211–212
Dictionary member, 283
DictionaryK, V type, 337–339
  index initializers, 338
DictionaryBase, 282–283, 317
digit separators, 44–45
directives
  #else, 585
  #endif, 585
  #if, 585
Directory class, 642
  CreateDirectory( ) method, 643
Delete( ) method, 643
EnumerateDirectories( ) method, 643
EnumerateFiles( ) method, 643
EnumerateFileSystemEntries( ) method, 644
GetObject( ) method, 643
GetFiles( ) method, 643
GetFileSystemEntries( ) method, 643
Move( ) method, 644
directory path names, 646–647
DirectoryInfo class, 642, 646
Disconnect( ) method, 369–370
DisplayMessage( ) method, 371
DLR (Dynamic Language Runtime), 398
do loops, 73–75
documents, XML
  creating, 676–677
  looping through nodes, 680–683
domain model, refactoring, 504–511
dynamic keyword, 399
dynamic lookup, 398–402
dynamic type, 399–402
dynamic variables, 398–402

ecosystems, cross-platform programs and, 581
ElementName property, 468
#else directive, 585
Elvis operator, 324
#endif directive, 585
#region keyword, 38
Entity Framework, 732
Entity Framework Core library, 11
enum keyword, 92
EnumerateDirectories( ) method, 643
EnumerateFiles( ) method, 643
EnumerateFileSystemEntries( ) method, 644
enumerations, 83, 91–95
defining, 92–93
underlying type, 92
Equals( ) method, 397
error handling, 167–168
catch blocks, 168–175
finally blocks, 168–175
SEH (structured exception handling), 168
try blocks, 168–175
Error List window, 20, 25–26
errors
  fatal, 149
  logic, 149
  semantic, 149
  syntax, 149
escape sequence, 42, 45
event handlers, 363, 365–367
anonymous methods, 375
CardLib, 375–383
delegates, 363
multipurpose, 371–374
Options window (KarliCards), 465–467
return values, 374
WPF, 437–438
ErrorHandler type, 374
EventHandlerT type, 374
events, 195, 363
defining, 368–371
event-driven applications, 195
KeyDown, 441
raising, 363
subscribing to, 363–364
throwing, 363
WPF, 436–441, 483–484
exception classes, 361
exception filtering, 169
exception handling, 170–176
exceptions
custom, 361–363
namespaces and, 168, 176
explicit conversion, 84, 86–91
expression filter, 103
expression trees, lambda expressions as, 416
expression-bodied methods, 339
expressions
lambda expressions, 409–419
looping and, 48
operands, 46
operators, 46
query expressions, 706
fatal errors, 149
fields, 239–241
defining, 234–235
objects, 182–183
FileInfo class, 642
Copy( ) method, 643
Create( ) method, 643
Delete( ) method, 643
Move( ) method, 643
Open( ) method, 643
file system, monitoring, 664–669
file system access classes
Directory, 642
DirectoryInfo, 642
File, 642
FileInfo, 642
FileSystemInfo, 642
FileSystemWatcher, 642
Path, 642
FileStream class, 648
file position and, 650
FileAccess enumeration, 648–649
 FileMode enumeration, 648–649
GetBytes( ) method, 655
reading data, 650–653
Seek( ) method, 650
Write( ) method, 655
writing data, random access files, 653–655
FileSystemInfo class, 642
Attributes property, 645
CreationTime property, 645
CreationTimeUtc property, 645
Extension property, 645
FullName property, 645
LastAccessTime property, 645
LastAccessTimeUtc property, 645
LastWriteTime property, 645
LastWriteTimeUtc property, 645
Name property, 645
FileSystemWatcher class, 642, 664
monitoring system, 665–669
properties, 665
filtering, exception filtering, 169
Finalize( ) method, 212
finally blocks, 168–175
floating-point coordinates, vector graphics and, 426
floating-point variables, 40
flow
branching, 59
looping, 59
for loops, 77–78
foreach loops, 101, 297
iterators, 285–288
LINQ, 708
forking
BCL (Base Class Libraries) and, 583
cross-platform programs and, 582
.NET Microframework, 584
FrameworkPropertyMetadata constructor, 487–488
frameworks, cross-platform programs and, 581
functions, 118
defining, 118–120
dele tates, 144–147
local, 137–138
Main( ), 138–141
overloading, 142–143
parameters, 122–124
arrays, 124–126
global data and, 136–137
matching, 124
out parameters, 129–130
reference parameters, 126–128
tuples, 130–131
value parameters, 126–128
return values, 120–122
global data and, 136–137
reusable code and, 118
signature, 118
struct functions, 141–142
ThrowException( ), 173–174
using, 118–120

G

global data, 136–137
global keyword, : : operator, 361
global names places, 54–56
global variables, 133
Grid control, 441
Grid Control, 44
Group Name property, 461
GZipStream class, 642

H

hardware platform, cross-platform programs and, 581
Health Endpoint Monitoring cloud programming pattern, 538
Hejlsberg, Anders, 360
hierarchy, namespaces, aliases, 360–361
Hierarchy window, 265–266
HorizontalAlignment property, 442
HTML, ASP.NET controls and, 609
HttpClient( ) method, 585
hybrid cloud, 534–535

I

IaaS (Infrastructure as a Service), Azure VM, 538
ICollection interface, 270, 276
IComparable interface, 308
IComparer interface, 308
IDEs (Integrated Development Environments), 11, 13, 32
IDictionary interface, 270, 282–283, 317
IEnumer able interface, 270, 276, 282–284
#if directive, 585
if statement, 65–69
get keyword, 236
GetBytes( ) method, 655
GetCopy( ) method, 289
GetDirectories( ) method, 643
GetEnum erator( ) method, 274
GetFiles( ) method, 643
GetFile SystemEntries( ) method, 643
GetHashCode( ) method, 302, 397
GetType( ) method, 211
Git repositories, Web App deployment and, 564–567
GitHub
CORECLR repository, 592
COREFX repository, 592
cross-platform programs and, 582
global data, 136–137
global keyword, : : operator, 361
global namespaces, 54–56
global variables, 133
Grid control, 441
Group Name property, 461
GZipStream class, 642
IList interface, 270
   CollectionBase class, 276
implicit conversion, 48, 84–86
implicit type, 392–393
IMyInterface, 296
increment operators, 48
index initializers, 338
indexers, 277–278
   key-accessed, 282–283
infinite loops, 79–80
inheritance, 188–189
   abstract classes, 190
   base classes, 188
   derived classes, 189
generic classes, 348–349
   sealed classes, 191
UML, 187
initializers, 386, 390–391
   collection initializers, 389–390
   object initializers, 387–389
InsertAfter( ) method, 685
InsertBefore( ) method, 685
instance constructors, 186
instances, 181
   classes, static members, 185–186
int arrays, 124
int integer type, 39
integer types
   byte, 39
   int, 39
   long, 39
   sbyte, 39
   short, 39
   uint, 39
   ulong, 39
   ushort, 39
interface keyword, 206–207
interfaces
   abstract classes and, 226–228
   declaring, 206–207
   generic, defining, 350
ICollection, 270
IComparable, 308
IComparer, 308
IDictionary, 270
IEnumerable, 270
IList, 270
   implementing, 249–252
   member implementation, explicit, 251–252
objects, 187–188
System.Collections namespace, 270
internal classes, 204
internal members, 234
interrupting loops, 78–79
I/O (input/output), 642
is operator, 294–297
IsChecked property, 460, 461
IsEditable property, 461–462
IsEnabled property, 459, 468
IsReadOnly property, 461–462
IsThreeState property, 460
iterators, 283–285, 317
   collections and, 288
   implementing, 285–288
JavaScript, ScriptObject type and, 398
JIT (just-in-time) compiler, 6, 8, 580
JSON (JavaScript Object Notation), 674–675
   conversion from XML, 689–691
KarliCards, 458, 519–527
   About window, 455–458
   CardsInHand control, dependency properties, 520
   ComputerPlayer class, 504–511
   Deck class, InsertAllCards( ) method, 510
domain model, refactoring, 504–511
enumerations
   ComputerSkillLevel, 505
   PlayerState, 505
   GameDecks control, 519
dependency properties, 523
ListBox control, 472–473
Main window, 499–504
Options window, 458
   CheckBox control, 459–460
   ComboBox control, 461–462
designing, 463–465
event handling, 465–467
RadioButton control, 460–461
TabControl control, 462–463
TextBox control, 459
   Player class, 504–511
StartGameWindow, 473–476
KarliCards (continued)
user controls, 489–499
view model, 511–519
key values
collections, 282–283
KeyDown event, 441
keywords
# and, 38
abstract, 204, 250–251
async, 168
await, 168
base, 214, 246
checked, 314
class, 204–206
default, generic class definition, 342
delegate, 410
dynamic, 399
#endregion, 38
enum, 92
get, 236
global, 361
interface, 206–207
internal, 234
namespace, 43
new, 249–250
operator, 299
partial, 253
private, 234
protected, 234
public, 234
#region, 38
return, 121
sealed, 204
set, 236
static, 120
struct, 96–97
this, 215, 246–247, 277–278
using, 43, 188
using static, 55
var, 393, 395, 706–707
virtual, 250–251
void, 120

collections and, 416–419
as delegates, 415–416
as expression trees, 416
LINQ, 710–712
parameters, 413–415
statement bodies, 414–415
languages
data languages, 673
JSON, 674–675
XML, 674
layout, 441
alignment, 442
Border property, 443
columns, 451–452
dimensions, 442
margins, 442
padding, 442
panels, 444
Canvas control, 444–446
DockPanel control, 444, 446–448
Grid controls, 449–452
StackPanel control, 444, 448–449
WrapPanel control, 444, 449
rows, 451–452
stack order, 442
libraries
class libraries, 222–226
cross-platform programs and, 581
Entity Framework Core library, 11
linking, 9
LINQ (Language Integrated Query), 11, 409, 697–698
aggregate operators, 717–720
clauses, 707–708
explicit syntax, 709
extension methods, 709
foreach statement, 708
joins, 727–728
lambda expressions, 710–712
method syntax, 709–712
orderby clause, 713–714
ordering by multiple levels, 723–725
queries
deferred execution, 708–709
group queries, 725–726
large data sets, 714–716
results ordering, 712–713
SELECT DISTINCT, 720–723
query syntax, 705–706
method syntax and, 709–710

Label control, 430
lambda expressions, 409
anonymous methods, 410–413
restricted operators, 708
variables, var keyword, 706–707
LINQ providers
LINQ to Data Set, 704
LINQ to Entities, 704
LINQ to JSON, 704
LINQ to Objects, 704
LINQ to SQL, 704
LINQ to XML, 704
PLINQ, 704
LINQ to XML
fragments, 700–704
functional construction, 698–701
Linux, 4
ListBox control, KarliCards
SelectedIndex property, 473
SelectedItem property, 473
SelectedItems property, 473
SelectionMode property, 472
literal values, 43–45
literals, 42
string literals, 45–46
LoadCompressedFile() method, 664
local functions, 137–138
local variables, 133
logic errors, 149
long integer type, 39
looping, 59, 72
do loops, 73–75
foreach loops, 101, 708
iterators, 285–288
infinite loops, 79–80
interrupting loops, 78–79
for loops, 77–78
while loops, 75–77

Mac OS, 4
Main() function, 138–141
Main window (KarliCards), 481–483, 499–504
managed code, 7
Margin property, 442
matching parameters, 124
Materialized View cloud programming pattern, 537
mathematical operators
%, 47
*, 47
+, 47
-, 47
/, 47
decrement, 48
increment, 48
string concatenation, 48
variables and, 47–52
Max() method, 717
members
interfaces, explicit implementation, 251–252
internal, 234
private, 234
protected, 234
public, 234
refactoring, 242–243
MemberwiseClone() method, 289
Menu control, KarliCards, 499
MessageArrivedEventArgs class, 372
MessageHandler delegate type, 370
metadata, 7
cross-platform programs and, 582
metapackages, cross-platform programs and, 582
method groups, 337
method parameters, 402
methods, 239–241
Add(), 283
AddRange(), 276
Aggregate(), 416, 418
anonymous, 375, 409–413
Append(), 335
AppendChild(), 685
AppendFormat(), 335
Average(), 717
base classes, hiding, 244–246
ChecForMessage(), 370
Clear(), 276
Clone(), 289
Compare(), 308, 335
CompareTo(), 308
Connect(), 369–370
Copy(), 643
Count(), 717
Create(), 643
CreateAttribute(), 685
CreateComment(), 685
CreateDirectory(), 643
CreateElement(), 685
CreateNode(), 685
CreateTextNode(), 685
Deconstruct(), 242
methods (continued)
  defining, 235–236
    partial definitions, 253–255
Delete( ), 643
Disconnect( ), 369–370
DisplayMessage( ), 371
EnumerateDirectories( ), 643
EnumerateFiles( ), 643
EnumerateFileSystemEntries( ), 644
Equals( ), 397
Finalize( ), 212
generic, 347
  defining, 351–352
GetBytes( ), 655
GetCopy( ), 289
GetDirectories( ), 643
GetEnumerator( ), 274
GetFiles( ), 643
GetFileSystemEntries( ), 643
GetHashCode( ), 302, 397
GetType( ), 211
HttpClient( ), 585
InsertAfter( ), 685
InsertBefore( ), 685
LoadCompressedFile( ), 664
Max( ), 717
MemberwiseClone( ), 289
Min( ), 717
Move( ), 643, 644
MoveNext( ), 284
objects, 183
Open( ), 643
PerformOperations( ), 412
ReadKey( ), 219
ReadLine( ), 659
ReadToEnd( ), 660
Register( ), 486
RemoteCall( ), 402
Remove( ), 283
RemoveAll( ) method, 688
RemoveAt( ), 276
RemoveChild( ) method, 688
SaveCompressedFile( ), 664
SelectNodes( ), 689
SelectSingleNode( ), 689
SimpleList( ), 285
Sum( ), 717
System.Object, 209–211
ToString( ), 211, 397

WebClient( ), 585
Write( ), 655
WriteChar( ), 367, 409
WriteLine( ), 38, 55
XAttribute( ), 700
XDocument( ), 700
XElement( ), 700
Min( ) method, 717
Mono, 4
Move( ) method, 643, 644
MoveNext( ) method, 284
MSBUILD (Microsoft Build Engine), 601
MSIL (Microsoft Intermediate Language), 6
MTOM (Message Transmission Optimization Mechanism), 763
multidimensional arrays, 106–108
multipurpose event handlers, 371–374
MVC (Model-View-Controller), 612

N

named parameters, 404–408
namespace keyword, 43
namespace statement, 37
namespaces, 54–56
  : : operator, 360–361
exceptions and, 168, 176
global namespace, 54
hierarchy, aliases, 360–361
nested type definitions, 247–249
System.Collections, 270
System.Collections.Generic, 329–332
System.Diagnostics, 152, 155
System.IO, 642
System.Text, 329
System.Threading.Tasks, 329
XAML, 428–429
naming, variables, 43
native code, 6
  execution, 9
nested type definitions, 247–249
.NET Compact Framework, 5
.NET Core, 13
  application local frameworks, 595–596
  applications, 13
    writing, 5–9
Console App, 602–605
Console App creation, 593–595
CORECLR repository (GitHub), 592
COREFX repository (GitHub), 592
cross platform, 590
modular design and, 593
NuGet package, 593
installation, 593–595
open source, 591
optimization for cloud and, 591–592
performance versus .NET Framework, 592
porting to from .NET Framework
feature availability, 606
.NET Framework target upgrade, 606
platforms targeted, 606–607
SDK installation, 588–590
self-contained deployment, 595–596
.NET Framework, 4, 13, 579
applications, 13
writing, 5–9
BCL (Base Class Libraries), 5
C#, 4
CIL (Common Intermediate Language), 580
AOT (Ahead of Time), 580
JIT (Just in Time), 580
cloud and, 536
CLR (Common Language Runtime), 5
consumption, languages, 580
CTS (Common Type System), 5
library, types, 5
MSIL (Microsoft Intermediate Language), 580
OOP (object-oriented programming), 4
porting to .NET Core from, 605
API Portability Analyzer, 605
feature availability, 606
.NET Framework target upgrade, 606
platforms targeted, 606–607
third-party dependencies and, 605–606
.NET Micro Framework, 5
.NET Microframework, forking, 584
.NET Native, cross-platform programs and, 583
.NET Standard, 583
API count, 587
class library
creating, 596–599
packaging, 599–602
frameworks supported, 587
namespaces, 587
PCLs (Portable Class Libraries) and, 586
platforms supported, 587
shared projects, 584
new keyword, 249–250
NGEN, AOT compiler and, 580
NuGet
cross-platform programs and, 582
.NET Core and, 593
null coalescing (??) operator, 323–324
null condition (?.) operator, 325
nullable types, 196, 321, 325–329
operators and, 322–323
null-coalescing operator, 175

Object Browser, 218–219
object initializers, 387–389
object-relational mapping, 732
objects
classes, 181
construction, 184
constructors, 184–185
destructors, 185
fields, 182–183
methods, 183
properties, 182–183
timer, 366
UML and, 181
OOP (object-oriented programming), 4,
179–180
classes
collections, 194
containment, 193–194
desktop applications, 196–199
events, 195
inheritance, 188–191
interfaces, 187–188
operators, overloading, 194–195
overview, 180–181
polymorphism, 191–192
interface polymorphism, 192–193
Open( ) method, 643
open source
cross-platform programs and, 581
.NET Core, 591
operands, expressions, 46
operator keyword, 299
operator overloading
operator overloading (continued)
CarLib, 302–307
syntax, 298–299
value comparisons and, 298–302
operators
as, 315–316
?? (null coalescing), 323–324
::, 360–361
==, 397
assignment, 42
?. (null condition) operator, 325
aggregate, LINQ, 717–720
assignment, 52–53
binary, overloading, 301
Boolean, 60–62
comparison, overloading, 301
conditional, 47
expressions, 46
genetic, 349–350
is, 294–296
is pattern matching, 297
nullable types and, 322–323
overloading, 194–195
conversion operators, 313–314
precedence, 53–54, 64–65
unary, overloading, 301
optional parameters, 403, 405–408
OptionalAttribute attribute, 404
order, 404
values, 404
OptionalAttribute attribute, 404
Options window (KarliCards), 458
CheckBox control, 459–460
ComboBox control, 461–462
data binding
DataContext control, 468
dynamic binding, 470–472
local objects, 468
static binding, external objects, 469
designing, 463–465
event handling, 465–467
RadioButton control, 460–461
TabControl control, 462–463
TextBox control, 459
orientation enumeration type, 94, 98
out parameters, 129–130
overflow checking, 87–88
overloading functions, 142–143
overloading operators, 194–195
>, 301
!=, 302
==, 302
binary, 301
CardLib, 302–307
comparison, 301
conversion operators, 313–314
unary, 301
value comparisons and, 298–302
overriding methods, calling, 246–247

PaaS (Platform as a Service), Azure Cloud Services, 538
packages, cross-platform programs and, 582
Padding property, 442
Panel class, 444
Canvas control, 444–446
DockPanel control, 444, 446–448
Grid controls, 449–452
StackPanel control, 444, 448–449
WrapPanel control, 444, 449
parameters, 122–124
arrays, 124–126
global data and, 136–137
lambda expressions, 413–415
matching, 124
method parameters, 402
named, 404–408
optional, 403, 405–408
OptionalAttribute attribute, 404
order, 404
values, 404
out parameters, 129–130
reference parameters, 126–128
tuples, 130–131
value parameters, 126–128
partial class definitions, 252–253
partial keyword, 253
partial method defintion, 253–255
Path class, 642
path names
absolute, 646–647
relative, 647
pattern matching
const pattern, 297
is operator, 297
switch case statement and, 297
switch statement and, 102–103
PCLs – ScriptObject type

type pattern, 297
var pattern, 297
PCLs (Portable Class Libraries)
creating, 585–586
cross-platform programs and, 582
shared projects, 584
PerformOperations( ) method, 412
Player class, 504–511
pointers, 230
polymorphism, 191–192
covariance and, 353
contravariance and, 353
interface polymorphism, 192–193
precedence of operators, 53–54, 64–65
private cloud, 534–535
private members, 234
properties, 239–241
accessors, 236, 252
alignment, 442
AllowMultiple, 385–386
automatic, 243–244
CheckBox control, 460
ComboBox control, 461–462
Content, 441
defining, 236–241
indexers, 277–278
objects, 182–183
RadioButton control, 461
read-only, 395
TextBox control, 459
Value, 323
WPF, dependency property, 431
XAML
attached properties, 435–436
dependency properties, 435
Properties window, 20, 25
protected members, 234
public cloud, 534–535
public members, 234

query expressions, 706

RadioButton control, 461
random access files, 650

Razor, 548, 551, 553
reading attributes, reflection, 384–385
ReadKey( ) method, 219
ReadLine( ) method, 659
read-only properties, 395
ReadToEnd( ) method, 660
refactoring, 242–243
reference parameters, 126–128
reference types, 195–196, 230
reflection, attributes and, 384–385
#region keyword, 38
Register( ) method, 486
relational databases, 731. See also SQL (Structured Query Language)
relational operators, Boolean logic, 60
relative path names, 647
RemoteCall( ) method, 402
remoting, 761–762
Remove( ) method, 283
RemoveAll( ) method, 688
RemoveAt( ) method, 276
RemoveChild( ) method, 688
rendering
2D, 426
3D, 426
resources, CIL, 7
REST (Representational State Transfer), WCF and, 762
RESTful HTTP services, 10
Retry cloud programming pattern, 538
return keyword, 121
return values, 120–122
event handlers, 374
global data and, 136–137
reusable code, functions and, 118
route struct, 98
routed commands, 438–439
KarliCards, 499–504
routed events, 437–441
row layout, 451–452
runtime, 7
cross-platform programs and, 582

SaveCompressedFile( ) method, 664
sbyte integer type, 39
ScriptObject type, 398
sealed classes, 204
sealed keyword, 204
searching, generic types, 331–337
SEH (structured exception handling), 168
SelectedIndex property, 462
SelectedItem property, 462
SelectNodes( ), 689
SelectSingleNode( ), 689
semantic errors, 149
semantic versioning, 594
cross-platform programs and, 582
serial devices, streams and, 647
set keyword, 236
shallow copies, 230–231
Sharding cloud programming pattern, 537
shared projects
.NET Standard and, 584
PCL (Portable Class Libraries) and, 584
short integer type, 39
simple variable types, 38–42
SimpleList( ) method, 285
SOA (service-oriented architecture), 762
SOAP (Simple Object Access Protocol), WCF and, 762
software platform, cross-platform programs and, 581
Solution Explorer window, 20
Class View, 24
XAML and, 429–430
sorting
collections, Ch11Ex05, 310–313
generic types, 331–337
SQL (Structured Query Language), 731–732
SQL Azure, 538
SQL Server Express, installation, 732
stacks, cross-platform programs and, 581
standards, cross-platform programs and, 581
StartGameWindow (KarliCards), 473–476
statements, 34
if, 65–69
namespace, 37
statement bodies, lambda expressions, 414–415
switch, 69–72
switch case, 297
using, 37, 360, 401
static classes, 186
static keyword, 120
storage, Azure storage account, creating, 539–540
storyboards, animations, 484–485
StreamReader class, 648, 658–660
streams
classes, 647–648
FileStream, 648–655
StreamReader, 648, 658–660
StreamWriter, 648, 655–657
serial devices and, 647
StreamWriter class, 648, 655–657
string concatenation operators, 48
String Interpolation, 42
string literals, 42, 45–46
string type, 40
strings, 109–114
struct keyword, 96–97
structs, 84, 97–98
versus classes, 228–230
defining, 96
functions, 141–142
route, 98
structured exception handling (SEH), 168
subscribing to events, 363–364
Sum( ) method, 717
switch case statement, 297
switch statement, 69–72
pattern matching and, 102–103
symbolic information, Debug (VS), 150
syntax
{} (curly brackets), 34
< > angle brackets, 320
/* with comments, 35
// with comments, 36
blocks, 34
case sensitivity, 36
comments, 35
elements, 149
statements, 34
whitespace characters, 34
System.Attribute class, 386–387
System.Collections namespace, 270
System.Collections.Generic namespace, 329–332
System.Diagnostics namespace, 152, 155
System.Exception class, 361
System.IO namespace, 642
System.IO.Compression namespace
DeflateStream class, 642
GZipStream class, 642
System.Object, methods, 209–211
System.Text namespace, 329
System.Threading.Tasks namespace, 329
System.Timers.ElapsedEventArgs, 366
System.Timers.ElapsedEventHandler, 366
System.Timers.Timer class, 366

**T**

target framework, cross-platform programs and, 582
Team Explorer, 20
Text property, 459, 462
TextBox control, 430, 459
TextWrapping property, 459
TFM (Target Framework Moniker), cross-platform programs and, 582
this keyword, 215, 246–247, 277–278
Throttling cloud programming pattern, 538
throw exceptions, 175
ThrowException( ) function, 173–174
timer object, 366
Toolbox, 19
ToString( ) method, 211, 397
TriggerBase class, 483–484
try blocks, 168–175
tunneling events, 438
tuples, 130–131
deconstructing, 241–242
type, implicit typing, 392–393
type comparisons, 317
boxing, 292–294
is operator, 294–296
unboxing, 292–294
type conversion, 83
explicit, 84, 86–91
implicit, 84–85
type definitions, nested, 247–249
type inference, 392–394
type patterns, 297
typeof( ) operator, 292
types
anonymous, 394–398
Boolean, 40
dynamic, 399–402
floating-point
decimal, 40
double, 40
float, 40
generic, 321
integers
byte, 39
int, 39
long, 39
sbyte, 39
short, 39
uint, 39
ulong, 39
ushort, 39
nullable, 196, 321, 325–329
operators and, 322–323
reference, 195–196, 230
text, 40
value, 195–196, 230
variables, simple, 38–42
typesafe language, 10

**U**

uint integer type, 39
ulong integer type, 39
UML (Unified Modeling Language), 181
unary operators, overloading, 301
unsigned variables, 40
URIs (Uniform Resource Identifiers), 428
ushort integer type, 39
using keyword, 43, 188
using statement, 37, 360, 401
attributes, 384
using static keyword, 55
UWA (Universal Windows App), 609
UWP (Windows Universal Platform), 793
adaptive displays, 797–800
adaptive triggers, 798–799
FlipView, 800–805
relative panel, 798
applications, 4
badges, 796–797
creating, 794–795
disk access, 807–810
lifetime, 797
Lock Screen, 797
menus, 796
navigation, 810–812
sandboxed, 805–807
screen orientation, 796
tiles, 796–797
toolbars, 796
CommandBar control, 812–814
state, managing, 812–816
XAML, 797
Valet Key cloud programming pattern, 537
value comparisons, 317
operator overloading, 298–302
value parameters, 126–128
Value property, 323
value types, 195–196, 230
values
optional parameters, 404
return values, event handlers, 374
variables, literals, 42
var keyword, 393, 395, 706–707
var pattern, 297
variables
arrays, 84, 99, 100–101
declaring, 99–100
foreach loops, 101
multidimensional, 106–108
assignment operators, 52–53
declaring, 38
dynamic, 398–402
enumerations, 83, 91–95
global, 133
LINQ, var keyword, 706–707
local, 133
mathematical operators, 47–52
naming, 38, 43
scope, 131–134
strings, 109–114
structs, 84, 96–98
types, 38
assigning, 38
Boolean, 40
floating-point, 40
integers, 39–40
simple, 38–42
text, 40
unsigned, 40
values, literals, 42
variance
contravariance, 353, 354–355
covariance, 353, 354
Vector class, 328–329
vector graphics, floating-point coordinates, 426
Vectors collection

Compare( ) method, 335
Sum( ) method, 335
versions, cross-platform programs and, 582
VerticalAlignment property, 442
verticals, cross-platform programs and, 581
VerticalScrollBarVisibility property, 459
view model, KarliCards, 511–519
virtual keyword, 250–251
Visual Debugging tools, 443–444
Visual Studio 2017, 3
debugging in, 22–23, 150
development environment, 16–20
Properties, 25
settings, 32
Solution Explorer, 24–25
solutions, 12
projects, 12
XAML View, 429
Visual Studio Code, 11
Visual Studio Community, 11, 590
Visual Studio Enterprise, 11, 12
Visual Studio Professional, 11, 12
void keyword, 120
void methods, 370

W
WAS (Windows Activation Service), 768
WCF (Windows Communication Foundation), 761
addresses, 764
behaviors, 767
bindings, 762, 765–766
contracts, 762, 766–767, 777–784
hosting, 768
HTTP, 763
addresses, 764
message patterns, 764
MSMQ and, 764
MTOM (Message Transmission Optimization Mechanism) and, 763
named pipes, 764
operations, 762
overview, 762–763
programming, 768–774
REST (Representational State Transfer) messages, 762
services, 11
  applications, 11
  client creation, 768–774
  self-hosted, 784–790
SOAP (Simple Object Access Protocol) messages, 762
TCP, 763
  addresses, 764
test client, 774–776
UDP, 764
  addresses, 764
Web APIs, 10
web applications, 10, 610–611
web services, 761
WebClient( ) method, 585
while loops, 75–77
whitespace characters, 34
windows, controls, 431–432
Windows Mobile, 4
Windows Store, applications, 10
  Badges, 816–818
  packaging, 818–819
  Tiles, 816–818
WindowsAzure.Storage library, 542
wizards, Add New Item, 219–220
working directory, 645
WPF (Windows Presentation Foundation), 426, 609.
  See also KarliCards; XAML
  animations, 484–485
  applications, 10
  Ch14Ex01, 429–430
  classes, dependency properties, 486–499
  control layout, 441–443
  controls, 430–432, 441, 481
debug mode, 443–444
desktop applications, 16
events, 436
  bubbling, 438
  control events, 436–437
  handling, 437–438
routed, 438, 439–441
  triggers, 483–484
  tunneling, 438
Main window (KarliCards), 481–483
properties, dependency property, 431
storyboards, 484–485
UserControl class, 485
WPF App project template, 429
XAML, 426–427
Write( ) method, 655
WriteChar( ) method, 367, 409
WriteLine( ) method, 38, 55
WSDL (Web Service Description Language), 762

X-Y-Z

XAML (eXtensible Application Markup Language),
  29–31, 426, 427–428
animation storyboarding, 427
code-behind files, 427, 429
text processing, 427
namespaces, 428–429
properties, 432–436
rendering
  2D, 426
  3D, 426
reusable resources, 427
routed commands, 438–439
Solution Explorer and, 429–430
UI objects, 427
UWP app development and, 797
XML and, 426
XAML View, 429
XAttribute( ) method, 700
XDocument( ) method, 700
XElement( ) method, 700
XML (eXtensible Markup Language), 426, 427
databases and, 750–756
documents
  creating, 676–677
  looping through nodes, 680–683
JSON, conversion to, 689–691
LINQ to XML
  fragments, 700–704
  functional construction, 698–701
nodes
  creating, 684–687
deleting, 687–689
  inserting, 685–686
  selecting, 689
value change, 683–684
XML (continued)
overview, 674
schemas, 675
XPath and, 691–695

XML DOM (XML Document Object Model), classes
XmlAttribute, 678
XmlComment, 678
XmlDocument, 677, 678
XmlElement, 677, 678–679
XmlNode, 677
XmlNodeList, 678

XmlText, 678
XmlAttribute class, 678
XmlComment class, 678
XmlDocument class, 677, 678
XmlElement class, 677
properties, 679–680
XmlNode class, 677
XmlNodeList class, 678
XmlText class, 678
XPath, 691–695