Index

! (logical negation), 57, 58
!= (not equal to), 57, 60
   for nullable types, 114
   for strings, 16
\ (double quote escape code), 16
\% (remainder), 57, 58
\%= (compound assignment), 62
\& (bitwise and), 57, 60
\& (logical strict and), 57, 58
   nullable, 114
\&\& (logical and), 57, 58
\&= (compound assignment), 62
\\ (single quote escape code), 16
* (multiplication), 57
*= (compound assignment), 62
+ (addition), 57
+ (delegate combination), 57, 112
+ (enum addition), 57, 110
+ (string concatenation), 18, 57
   example, 17
++ (postincrement), 57, 58
++ (preincrement), 57, 58
+= (compound assignment), 57, 62
+= (event assignment), 52, 57
- (delegate removal), 57, 112
- (enum subtraction), 57, 110
- (minus sign), 57
- (subtraction), 57
-= (postdecrement), 57, 58
-= (predecrement), 57, 58
-= (compound assignment), 62
-= (event assignment), 52, 57
/ (division), 57, 58
/= (compound assignment), 62
; (semicolon), 84
   in delegate type declaration, 112
   misplaced (example), 91
< (less than), 57, 60
<< (left shift), 57, 60
<<= (compound assignment), 62
<= (less than or equal to), 57, 60
= (assignment), 57
== (equal to), 57, 60
   for nullable types, 114
   for strings, 16
> (greater than), 57, 60
>= (greater than or equal to), 57, 60
>> (right shift), 57, 60
>>= (compound assignment), 62
?: (conditional expression), 57, 62
?? (null-coalescing), 57, 114
@ (identifier prefix), 4
@"..." (verbatim string constant), 16
\\\ (backslash escape code), 16
^ (bitwise exclusive-or), 57, 60
^ (logical strict exclusive-or), 57, 58
^= (compound assignment), 62
\0 (format specification), 20
| (bitwise or), 57, 60
| (logical strict or), 57, 58
   nullable, 114
|= (compound assignment), 62
|| (logical or), 57, 58
- (bitwise complement), 57, 60
\a (alert escape code), 16
abbreviation for constructed type, 154
abrupt termination, 85
Abs method (Math), 124
absolute value, 124
   example, 63, 87
abstract
   class, 34
   method, 42
   property, 48
abstract
   class modifier, 34
   method modifier, 42
abstract override (method modifier), 42
Accept method (Socket), 142
access modifier, 34
accessible member, 34
accessor
get, 48
set, 48
\texttt{Acos} method (Math), 124
actual parameter. \textit{See} argument
\textit{actual-list}, 68
add clause, 52
\texttt{Add} method
\texttt{ICollection<T>}, 158
\texttt{IDictionary<K,V>}, 164
\texttt{IList<T>}, 164
addition operator (+), 57
\texttt{AddModule} (compile option), 2
\texttt{AddRange} method (\texttt{List<T>}), 166
\texttt{After} (iterator state), 100
ambiguous method call, 41, 72
amortized complexity, 159
and operator
\texttt{bitwise} (\&), 57
\texttt{logical} (\&\&), 57
\texttt{logical strict} (\&\&), 57
\texttt{nullable}, 114
anonymous method expression, 82
APoint class (example), 33, 67, 69
\texttt{Append} method (\texttt{StringBuilder}), 24
\texttt{AppendFormat} method (\texttt{StringBuilder}), 24
\texttt{AppendText} method (\texttt{FileInfo}), 140
applicable method, 72
ApplicationException, 116
arccosine \((\texttt{Acos})\), 124
arcsine \((\texttt{Asin})\), 124
arc tangent \((\texttt{Atan} \text{ or } \texttt{Atan2})\), 124
\texttt{args} (command line arguments), 2
argument
by-value, 70
of method call, 68
out, 70
ref, 70
ApplicationException, 116, 130, 132, 134, 136, 138–140, 158, 164
ArgumentOutOfRangeException, 18, 24, 116, 130, 132, 134, 136, 138, 139, 158, 164, 166
arithmetic
expressions, 58
operators, 58
array, 26–31
access, 26, 28, 57
of arrays, 28
assignment to element, 26
assignment type check, 26
creation expression, 26, 28
C-style, 28
element, 26
\texttt{initial value}, 26
type, 26
\texttt{implements} \texttt{IList<T>}, 88
index, 26
initialization, 26
is enumerable, 88
iteration (example), 31
jagged, 28
Java-style multi-dimensional, 28
length, 26
member, 30
multi-dimensional, 28
product (example), 63
rank, 26, 30
rectangular, 28
reversal (example), 89
search, 30
segment swapping (example), 31
sort, 30
type, 26
Array class, 8, 30
\texttt{not usable as constraint}, 148
ArrayList (example), 33, 137, 145
\texttt{ArgumentNullException}, 6, 30
\texttt{ASCII character encoding}, 16
\texttt{Asin} method (Math), 124
\texttt{AsReadOnly} method (\texttt{List<T>}), 166
assembly, 2
\texttt{reference}, 2
\texttt{Assert} method (Debug), 176
assertion, 176
assigned, definitely, 14
assignment
  array element, 26
  compound, 57, 62
  event, 52, 57
  expression, 62
  operators (+=, *...), 62
  statement, 85
  value type variable, 6
associativity, 56, 57
Atan method (Math), 124
Atan2 method (Math), 124
attribute, 178–179
  custom, 178
  predefined, 178
Attribute class, 178
AttributeUsage (attribute), 178
\b (backspace escape code), 16
backslash escape code (\), 16
backspace escape code (\b), 16
Bank class (example), 121
base
  class, 36
    constructor call, 36, 44
    constructor call (example), 75, 107
    immediate, 36
  interface, 108
base, 36
  base class constructor call, 36, 37
  base class field access, 37, 66
  base class indexer access, 37, 78
  base class method call, 37, 72
  base class property access, 37, 76
base-interfaces, 106
BaseStream property (BinaryWriter), 134
Before (iterator state), 100
best signature, 68
better
  conversion, 11
  signature, 68
  type, 68
binary
  input, 134
  operator overloading, 50
  output, 134
  stream, 126
BinaryFormatter class, 129, 179
BinaryReader class, 134
  example, 137
BinarySearch method
  Array, 30
  List<T>, 166
BinaryWriter class, 134
  example, 137
Bind method (Socket), 142
bitwise
  and operator (\&), 57, 60
  complement operator (~), 57, 60
  exclusive-or operator (^), 57, 60
  operators, 60
  or operator (|), 57, 60
block, 84
  statement, 84
    example, 87
block-statement, 84
blocking method call, 128
bool (simple type), 7
Boolean struct type (bool), 7
Box-Muller transformation (example), 125
boxing
  conversion, 12
  of value as object, 103, 104
breadth-first graph traversal (example), 171
break statement, 92
  example, 85
Buffer class (example), 123
by-value
  argument, 70
  parameter, 40, 70
byte (simple type), 7, 10
byte array
  reading from, 139
  writing to, 139
byte stream, 126
  reading from, 130
  writing to, 132
Byte struct type (byte), 7
bytecode, 2

C++ templates versus C# generics, 156
C-style multi-dimensional array, 28
calendar (example), 111
call
   of delegate, 112
   of method, 68–75
call-by-value, 70
CanRead property (Stream), 136
CanSeek property (Stream), 136
CanWrite property (Stream), 136
Capacity property
   List<T>, 166
   MemoryStream, 139
captured variable, 82
carriage return escape code (\r), 16
case clause, 86
cast. See type cast
catch clause, 96
catching an exception, 96
Ceiling method (Math), 124
class (simple type), 7, 10
Char struct type (char), 7
character
   counting (example), 17
   encoding, 130
      UTF-8, 130, 132
   escape code, 16
   stream, 126
type, 6
check date (example), 27
checked
   expression, 57, 58
   statement, 98
checked context, 58, 98
choice statement, 86
choosing a collection class, 159
class, 32–53
   abstract, 34
   base, 36
   body, 32
declaration, 32
   partial, 174
generic, 144, 146, 148
   hierarchy, 36
      example, 35
implementing interface, 108
inner (Java), 46
instance, 54
internal, 46
library, 172
loading, 2, 38, 44, 54
member, 32
modifier, 34
nested, 46
of an object, 54, 64
operator overloading, 50
public, 46
run-time representation, 54
sealed, 34
   example, 175
   static, 34
type parameter, 146
   versus struct type (example), 103
class constraint, 148
class-base-clause, 36, 108
class-body, 32
class-declaration, 32
class-modifier, 32
Clear method
   ICollection<T>, 158
client socket, 142
Clone method (String), 16
Close method, 128
   BinaryWriter, 134
   Socket, 142
   Stream, 136
   TextReader, 130
   TextWriter, 132
closure, 112
   for anonymous method, 82
code
   non-static, 32
   static, 32
collection, 158–171
Color enum type (example), 111
Colored interface (example), 107
ColoredDrawable interface (example), 107
ColoredDrawablePoint class (example), 107
ColoredPoint class (example), 107
ColoredRectangle class (example), 107
ColoredPoint class, generic (example), 147
command line arguments, 2
   example, 17, 25
comment, 4
Common Language Runtime, ix
comparable, 162–163
Compare method
   long (example), 63
   IComparer, 162
   IComparer<T>, 162
   String, 18
comparer, 162–163
CompareTo method
   IComparer, 162
   IComparer<T>, 162
   simple types, 162
   String, 18
comparison
   operator, 57, 60
   of references, 8
compatible
   anonymous method and delegate type, 82
   method and delegate type, 112
   operand type, 60
compile, 2
compile option, 2
   /addmodule, 2
   /d, 2, 176
   /debug, 2
   /define, 2
   /main, 2
   /r, 2
   /reference, 2
   /t, 2
   /target, 2
compile-time
   constant, 86
   type, 6, 56
compiler, 2
complement operator (−), 57
complete program (example), 3
complexity
   amortized, 159
   of collection class operations, 159
   time, 159
compound assignment, 57, 62
   to indexer, 78
   for nullable types, 114
   operators, 62
   to property, 76
Concat method (String), 18
concatenating arguments (example), 17, 25
concatenation of strings, 24
concurrency, 118–123
Conditional (attribute), 178
conditional expression (?:), 62
Console.Error (standard error), 132
Console.In (standard input), 130
Console.Out (standard output), 132
Console.WriteLine method, 132
const
   constant field declaration, 38
   local constant declaration, 84
constant, 38
   in class, 38
   compile-time, 86
   declaration, 38, 84
   floating-point, 7
   integer, 6
   in method, 84
   simple type, 6
   string, 16
constraint on type parameter, 148
constructed type, 144
abbreviation for, 154
constructor, 44
   body, 44
   call, 64
   base class, 44
   example, 75
   declaration, 44
   default, 36, 44
generic (no such thing), 144
overloading, 44
predefined in struct, 102
signature, 44
static, 44
constructor-declaration, 44
constructor-modifier, 44
Contains method
   ICollection<T>, 158
ContainsKey method (IDictionary<K,V>), 164
ContainsValue method
   Dictionary<K,V>, 168
continue statement, 92
   example, 85
conversion, 10–13
   between reference types, 12
      explicit, 10, 52
         user-defined, 52
      implicit, 10, 52
         user-defined, 52
   overflow checking, 10
   standard
      boxing, 12
      between simple types, 10
      unboxing, 12
      user-defined, 52
         example, 81
Copy method (String), 16
CopyTo method
   ICollection<T>, 158
   List<T>, 166
Cos method (Math), 124
Count property (ICollection<T>), 158
Create method (FileInfo), 140
Created (thread state), 119
CreateText method (FileInfo), 140
creation
   of array, 26, 28
   of delegate, 112
   of object, 64
   of stream, 126
   of struct value, 64
   of thread, 118
CryptoStream class, 126
.cs file, 2
   example, 175
csc (Microsoft C# compiler), 2
   example, 175
culture, 23
culture sensitive, 23
   string formatting, 20
CultureInfo class, 20
currency formatting as string, 22
current
   object, 32, 66
   thread, 20
Current property (IEnumerator<T>), 160
CurrentCulture property (Thread), 20
CurrentEncoding property (StreamReader), 130
CurrentThread method (Thread), 122
custom
   attribute, 178
   DateTime formatting, 22
   number formatting, 22
/d (compile option), 2
data member, 32
DataAvailable property (Socket), 142
database query (example), 167
Date class (example), 111
date formatting as string, 22
DateTime class, 22
DateTimeFormat property (CultureInfo), 20
day enum (example), 111
DComparer
   delegate type (example), 113
      interface, generic (example), 155
de-DE culture (example), 23
Dead (thread state), 119
/debug (compile option), 2
Debug class, 176
decimal (simple type), 6, 7, 10
decimal integer constant, 6
Decimal struct type (decimal), 7
declaration
   class, 32
      in class body, 32
      constant, 38, 84
constructor, 44
custom attribute, 178
delegate type, 112
decimal member, 110
decimal type, 110
event, 52
field, 38
formal parameter, 40
indexer, 48
interface, 106
local variable, 84
method, 40
namespace, 172
property, 48
statement, 84
struct type, 102
declared type, 6
decrement operator (\(\sim\)), 57, 58
default
clause in switch, 86
constructor, 36, 44
in struct type, 102
namespace, 172
value, 14
for simple type, 7
for struct type, 102
for type parameter, 145, 150
of field, 38
default\(t\) (default value for type \(t\)), 14
example, 145
#define (compile option), 2
definitely assigned, 14
delegate
call, 112
combination operator (\(\ast\)), 112
conversion from anonymous method, 82
creation, 112
generic, 154
invocation list, 112
removal operator (\(-\)), 112
return type, 112
type, 112–114
declaration, 112
generic, 154–155
Delegate class, 8, 112
not usable as constraint, 148
dehable-modifiers, 112
dehable-type-declaration, 112
Delete method (FileInfo), 140
deepth-first graph traversal (example), 171
Dequeue method (Queue\(<T>\)), 170
derived class, 36
Deserialized method
BinaryFormatter, 128
SoapFormatter, 128
deterministic finite automaton (example), 169
dictionary, 168–171
Dictionary\(<K,V>\) class, 168
die
de example, 27, 133
loaded (example), 43
Directory class, 140
directory hierarchy (example), 141
DirectoryInfo class, 140
DirectoryNotFoundException, 116, 130, 132, 140
Dispose method
IDisposable\(<T>\), 98
IEnumerator\(<T>\), 160
DivideByZeroException, 58, 105, 116
division
by zero, 58
floating-point, 58
integer, 58
operator (\(\ast\)), 57
.dll file, 2
do-while statement, 90
double (simple type), 7, 10, 124
double quote (\(*\) escape code (\"\), 16
Double struct type (double), 7
drawable interface (example), 107
dynamically enclosing, 94
E constant (Math), 124
efficiency
array access, 28
avoid boxing, 8
avoid repeated string concatenation, 17
avoid throwing exceptions, 94
comparisons, 162
generic collection class, 79
generics, 156
indexer accessors, 48
property accessors, 48
string builder editing, 25
string concatenation, 24, 25
use generic collections, 12
eight queens problem (example), 101
element of array, 26
element type of array, 26
else keyword, 86
empty statement, 84
en-US culture (example), 23
Enabled (thread state), 118, 119
Encoding class, 130
encoding of characters, 130, 132
Encoding property (TextWriter), 132
end-of-stream, 130
EndOfStreamException, 116, 128, 134
Enqueue method (Queue<T>), 170
entry point, 2
enum
  example, 61
  member
    access, 110
    declaration, 110
    type of, 110
    type, 110–111
    arithmetics, 57, 110
    declaration, 110
Enum class, 8
  not usable as constraint, 148
enum-type-declaration, 110
enumerable, 160–161
expression, 88
enumerator, 160–161
  example, 161
  object, 100
Epsilon constant (Double), 6
equal to operator (==), 57, 60
  for nullable types, 114
  for strings, 16
equality of strings (example), 17
Equals method
  and hash code, 8
  array, 30
  default for struct type, 8
  IComparable<T>, 162
  IComparer<T>, 162
  Object, 8
  String, 16
  ValueType, 8
escape code, 16
evaluation
  of expression, 56
  of method call, 70
  order, 56
event, 52
  assignment, 52, 57
  declaration, 52
  description, 106
  generic (no such thing), 144
  handling, 52
  event-description, 106
exception, 116–117
  catching, 96
  throwing, 94
  type, 116
Exception class, 94, 116
exchange rate array (example), 29
exclusive-or operator (^), 57
.exe file, 2
  example, 3, 175
executable, 2
execution, 2
Exists property (FileInfo), 140
Exp method (Math), 124
explicit
  conversion, 10
  example, 63, 81
  user-defined, 52
  interface member implementation, 40, 108
  explicit, 52
exponential notation, 22
expression, 56–83
<table>
<thead>
<tr>
<th>Index</th>
<th>191</th>
</tr>
</thead>
<tbody>
<tr>
<td>anonymous method, 82</td>
<td></td>
</tr>
<tr>
<td>arithmetic, 58</td>
<td></td>
</tr>
<tr>
<td>array access, 26, 28</td>
<td></td>
</tr>
<tr>
<td>array creation, 26, 28</td>
<td></td>
</tr>
<tr>
<td>assignment, 62</td>
<td></td>
</tr>
<tr>
<td>conditional, 62</td>
<td></td>
</tr>
<tr>
<td>enumerable, 88</td>
<td></td>
</tr>
<tr>
<td>evaluation, 56</td>
<td></td>
</tr>
<tr>
<td>field access, 66</td>
<td></td>
</tr>
<tr>
<td>indexer access, 78</td>
<td></td>
</tr>
<tr>
<td>logical, 58</td>
<td></td>
</tr>
<tr>
<td>method call, 68–75</td>
<td></td>
</tr>
<tr>
<td>object creation, 64</td>
<td></td>
</tr>
<tr>
<td>property access, 76</td>
<td></td>
</tr>
<tr>
<td>statement, 84</td>
<td></td>
</tr>
<tr>
<td>struct value creation, 64</td>
<td></td>
</tr>
<tr>
<td>type cast, 57, 80</td>
<td></td>
</tr>
<tr>
<td>Extension property (FileInfo), 140</td>
<td></td>
</tr>
<tr>
<td>external module, 2</td>
<td></td>
</tr>
<tr>
<td>\f (form feed escape code), 16</td>
<td></td>
</tr>
<tr>
<td>factorial (example), 125</td>
<td></td>
</tr>
<tr>
<td>field, 14, 38</td>
<td></td>
</tr>
<tr>
<td>access, 57, 66</td>
<td></td>
</tr>
<tr>
<td>declaration, 38</td>
<td></td>
</tr>
<tr>
<td>default value, 38</td>
<td></td>
</tr>
<tr>
<td>hiding, 38</td>
<td></td>
</tr>
<tr>
<td>initialization, 38</td>
<td></td>
</tr>
<tr>
<td>initializer, 38</td>
<td></td>
</tr>
<tr>
<td>instance field access, 66</td>
<td></td>
</tr>
<tr>
<td>modifier, 38</td>
<td></td>
</tr>
<tr>
<td>read-only, 38</td>
<td></td>
</tr>
<tr>
<td>example, 27, 51</td>
<td></td>
</tr>
<tr>
<td>shadowed, 14</td>
<td></td>
</tr>
<tr>
<td>example, 67</td>
<td></td>
</tr>
<tr>
<td>static, 38</td>
<td></td>
</tr>
<tr>
<td>access, 66</td>
<td></td>
</tr>
<tr>
<td>example, 43</td>
<td></td>
</tr>
<tr>
<td>field-declaration, 38</td>
<td></td>
</tr>
<tr>
<td>field-modifier, 38</td>
<td></td>
</tr>
<tr>
<td>file, 140</td>
<td></td>
</tr>
<tr>
<td>extension, 140</td>
<td></td>
</tr>
<tr>
<td>.cs, 2</td>
<td></td>
</tr>
<tr>
<td>.dll, 2</td>
<td></td>
</tr>
<tr>
<td>.exe, 2</td>
<td></td>
</tr>
<tr>
<td>.netmodule, 2</td>
<td></td>
</tr>
<tr>
<td>index (example), 165</td>
<td></td>
</tr>
<tr>
<td>name, 140</td>
<td></td>
</tr>
<tr>
<td>path (example), 141</td>
<td></td>
</tr>
<tr>
<td>reading from, 130, 138</td>
<td></td>
</tr>
<tr>
<td>example, 131</td>
<td></td>
</tr>
<tr>
<td>source, 2</td>
<td></td>
</tr>
<tr>
<td>writing to, 132, 138</td>
<td></td>
</tr>
<tr>
<td>example, 133</td>
<td></td>
</tr>
<tr>
<td>FileAccess enum type, 138</td>
<td></td>
</tr>
<tr>
<td>example, 61</td>
<td></td>
</tr>
<tr>
<td>FileInfo class, 140</td>
<td></td>
</tr>
<tr>
<td>FileMode enum type, 138</td>
<td></td>
</tr>
<tr>
<td>FileNotFoundException, 116, 130, 138</td>
<td></td>
</tr>
<tr>
<td>FileShare enum, 138</td>
<td></td>
</tr>
<tr>
<td>FileStream class, 138</td>
<td></td>
</tr>
<tr>
<td>example, 137</td>
<td></td>
</tr>
<tr>
<td>finally, 96</td>
<td></td>
</tr>
<tr>
<td>finite state machine (example), 95</td>
<td></td>
</tr>
<tr>
<td>fixed-point type, 6</td>
<td></td>
</tr>
<tr>
<td>Flags (attribute), 178</td>
<td></td>
</tr>
<tr>
<td>example, 61</td>
<td></td>
</tr>
<tr>
<td>float (simple type), 7, 10, 124</td>
<td></td>
</tr>
<tr>
<td>floating-point</td>
<td></td>
</tr>
<tr>
<td>constant, 7</td>
<td></td>
</tr>
<tr>
<td>division, 58</td>
<td></td>
</tr>
<tr>
<td>overflow, 58</td>
<td></td>
</tr>
<tr>
<td>remainder, 58</td>
<td></td>
</tr>
<tr>
<td>type, 6</td>
<td></td>
</tr>
<tr>
<td>Floor method (Math), 124</td>
<td></td>
</tr>
<tr>
<td>Flush method, 128</td>
<td></td>
</tr>
<tr>
<td>BinaryWriter, 134</td>
<td></td>
</tr>
<tr>
<td>Stream, 136</td>
<td></td>
</tr>
<tr>
<td>TextWriter, 132</td>
<td></td>
</tr>
<tr>
<td>fn (Standard ML language), 82</td>
<td></td>
</tr>
<tr>
<td>for statement, 88</td>
<td></td>
</tr>
<tr>
<td>example, 31</td>
<td></td>
</tr>
<tr>
<td>foreach statement, 88</td>
<td></td>
</tr>
<tr>
<td>example, 51</td>
<td></td>
</tr>
<tr>
<td>form feed escape code (\f), 16</td>
<td></td>
</tr>
<tr>
<td>formal parameter, 40</td>
<td></td>
</tr>
<tr>
<td>formal-list, 40</td>
<td></td>
</tr>
<tr>
<td>Format method (String), 20</td>
<td></td>
</tr>
</tbody>
</table>
format specification, 20
FormatException, 6, 20, 116
formatting
code, 22
data as string, 20–23
date as string, 22
number as string, 22
time as string, 22
Frac class (example), 53, 65
frame (in stack), 54
FullName property (FileInfo), 140
function
as delegate (example), 155
member, 32
garbage collector, 54
Gaussian random numbers (example), 125
generic
class, 144, 146–149
constructor (no such thing), 144
delegate type, 154–155
event (no such thing), 144
indexer (no such thing), 144
interface, 150–151
method, 152–153
operator (no such thing), 144
property (no such thing), 144
struct type, 156–157
generics, 144–157
versus C++ templates, 156
implementation, 156
versus Java language generics, 156
get-accessor
of indexer, 48
of property, 48
GetCurrentDirectory property (Directory), 140
GetDirectories method (DirectoryInfo), 140
GetEnumerator method
Array, 30
IEnumerator (example), 89
IEnumerator<T>, 160
GetHashCode method
IComparer<T>, 162
Object, 8
simple type, 8
String, 8
ValueType, 8
GetLength method (Array), 30
GetType method (Object), 8
GetValue method (Array), 30
goto statement, 94
element, 85
governing type, 86
graph traversal (example), 171
greater than operator (\(>\)), 57, 60
greater than or equal to operator (\(\geq\)), 57, 60
hash code, 8
and Equals method, 8
hash table. See dictionary
heap, 8, 54
hexadecimal integer constant, 6
example, 61
hiding
base class’s indexer (example), 79
field, 38
indexer, 36
member, 36
method, 36
horizontal tab escape code (\(\backslash t\)), 16
HTML output (example), 133
ICollection<T> interface, 158
IComparable interface, 162
IComparable<T> interface, 162
example, 149
IComparer interface, 162
IComparer<T> interface, 162
IEqualityComparer<key, value> class, 164
IDisposable interface, 98
example, 89, 161
IEEE754 floating-point standard, 6, 7, 124, 182
IEEE754remainder method (Math), 124
IEnumerator interface, 160
IEnumerator<T> interface, 160
IEnumerator<T> interface, 160
IEnumerable<T> interface, 160
if statement, 86
if-else statement, 86
IList<T> interface, 164
    implemented by array type T[], 88
illegal type cast, 80
implementation of interface, 108
implicit, 52
implicit conversion, 10
    example, 63
    user-defined, 52
IMyList interface (example), 151
increment operator (++), 57, 58
index
    into array, 26
    into string, 16
indexer, 48
    access, 57
        expression, 78–79
        compound assignment, 78
declaration, 48
description, 106
formal parameter, 48
generic (no such thing), 144
get-access, 78
hiding, 36
    example, 79
IDictionary<K,V>, 164
IList<T>, 164
overloading, 48
    example, 109
set-access, 78
String, 16
StringBuilder, 24
type, 48
indexer-description, 106
IndexOf method
    IList<T>, 164
    StringBuilder, 24
InsertRange method (List<T>), 166
instance
    field, 38
        access, 66
        member, 32
        method, 42
        of struct type, 54
        test expression, 64
        example, 65
int (simple type), 7, 10
Int16 struct type (short), 7
Int32 struct type (int), 7
Int64 struct type (long), 7
integer
    constant, 6
    division, 58
    overflow, 58
        checking, 58
        remainder, 58
        sequences (example), 47, 51, 101, 109
        square root (example), 177
        type, 6
interface, 106–110
    base, 108
of instance fields, 38, 44
of static fields, 38
of struct values, 102
initializer of field, 38
inner class (Java), 46
input, 126–144
    binary, 134
    byte-oriented, 136
    character-oriented, 130
    from keyboard (example), 119
    sequential, 130
input-output, 126–144
    byte stream, 136–139
    character stream, 130–133
    examples, 129
    socket, 142–144
    thread-safety, 128
Insert method
    IList<T>, 164
    StringBuilder, 24
IndexOutOfRangeException, 16, 24, 26, 97, 116
inefficiency. See efficiency
infinity constant, 6
inheritance, 36
    example, 35, 43
initialization
    of array, 26
    of instance fields, 38, 44
    of static fields, 38
    of struct values, 102
    initializer of field, 38
    inner class (Java), 46
    input, 126–144
        binary, 134
        byte-oriented, 136
        character-oriented, 130
        from keyboard (example), 119
        sequential, 130
    input-output, 126–144
        byte stream, 136–139
        character stream, 130–133
        examples, 129
        socket, 142–144
        thread-safety, 128
Insert method
    IList<T>, 164
    StringBuilder, 24
IndexOf method (List<T>), 166
instance
    field, 38
        access, 66
        member, 32
        method, 42
        of struct type, 54
        test expression, 64
        example, 65
int (simple type), 7, 10
Int16 struct type (short), 7
Int32 struct type (int), 7
Int64 struct type (long), 7
integer
    constant, 6
    division, 58
    overflow, 58
        checking, 58
        remainder, 58
        sequences (example), 47, 51, 101, 109
        square root (example), 177
        type, 6
interface, 106–110
    base, 108
of instance fields, 38, 44
of static fields, 38
of struct values, 102
initializer of field, 38
inner class (Java), 46
input, 126–144
    binary, 134
    byte-oriented, 136
    character-oriented, 130
    from keyboard (example), 119
    sequential, 130
input-output, 126–144
    byte stream, 136–139
    character stream, 130–133
    examples, 129
    socket, 142–144
    thread-safety, 128
Insert method
    IList<T>, 164
    StringBuilder, 24
IndexOf method (List<T>), 166
instance
    field, 38
        access, 66
        member, 32
        method, 42
        of struct type, 54
        test expression, 64
        example, 65
int (simple type), 7, 10
Int16 struct type (short), 7
Int32 struct type (int), 7
Int64 struct type (long), 7
integer
    constant, 6
    division, 58
    overflow, 58
        checking, 58
        remainder, 58
        sequences (example), 47, 51, 101, 109
        square root (example), 177
        type, 6
interface, 106–110
    base, 108
Index

declaration, 106
partial, 174
explicit member implementation, 108
generic, 150
implementation, 108
explicit, 40
member, 106
interface-declaration, 106
interface-modifier, 106
intermediate code, 2
internal
class, 46
member, 34
InternalBufferOverflowException, 116
Interrupt method (Thread), 122
intersection closure (example), 171
InvalidCastException, 10, 12, 80, 88, 116, 158
InvalidOperationException, 114, 116, 160, 170
InvalidProgramException, 116
invariant, 177
invocation list of delegate, 112
invocation of method. See method call
IOException, 116, 128, 130, 132, 134, 138
IPEndPoint, 142
IPrintable interface (example), 149
is (instance test), 57, 64
example, 103
ISeq interface (example), 109
ISO 8601 date format, 22
ISO week number (example), 111
IsReadOnly property (ICollection<T>), 158
iterating over array (example), 31
iterator
block, 100
method, 100
jagged array, 28
Java language, 180
differences to C#, 180–181
generics, versus C# generics, 156
inner class, 46
multi-dimensional array, 28
protected member, 34
Join method (Thread), 122
Joining (thread state), 118, 119
justification, 20
Key field (KeyValuePair<K,V>), 168
keyboard input (example), 119
Keys property (IDictionary<K,V>), 164
KeyValuePair<K,V> struct, 168
keyword, 4
   as identifier (example), 5
   recognizing (example), 169
label, 94
labeled statement, 94
lambda (Scheme language), 82
LastIndexOf method (List<T>), 166
layout of program, 4
leap year (example), 59
left associative, 56
left shift operator (<<), 57, 60
overloading, 50
length of array, 26
Length property
   Array, 30
   FileInfo, 140
   Stream, 136
   String, 16
   StringBuilder, 24
less than operator (<), 57, 60
less than or equal to operator (<=), 57, 60
lexicographic ordering
   pairs
      example, 149, 163
      strings, 18
library, 2
line counting (example), 127
LinkedList<T> class (example), 147
   enumerator, 161
list, 164–167
List<T> class, 166
Listen method (Socket), 142
loaded die (example), 43
loading of class, 2, 38, 44, 54
local variable declaration, 84
lock, 120
Lock method (FileStream), 138
lock statement, 98, 120
Locking (thread state), 118, 119
log
  example, 49
  generic (example), 145
  of pairs (example), 157
Log class (example), 49
Log method (Math), 124
Log10 method (Math), 124
logarithm, 124
logical
  and operator (&&), 57
  negation operator (!), 57
  operators, 58
  or operator (| |), 57
  strict and operator (&), 57
  strict exclusive-or operator (^), 57
  strict or operator (|), 57
long (simple type), 7, 10
LongComparer class (example), 63
loop statement, 88–91
  nested (example), 89
loss of precision, 10

/m (compile option), 2
machine model, 54–56
/main (compile option), 2
Main method, 2
manifest of assembly, 2
map, See dictionary
map function (example), 153
Math class, 124
mathematical functions, 124–125
Max method
  example, 41
  Math, 124
MaxValue constant, 6
  example, 59
mcs (Mono C# compiler), 2
member
  access modifier, 34
  accessible, 34
  data, 32
  explicit implementation of interface, 108
  function, 32
  hiding, 36
  instance, 32
  of simple type, 6
  static, 32
MemoryStream class, 139
Message property (Exception), 116
meta-data, 178
method, 40
  abstract, 42
  anonymous, 82
  body, 40
  call, 57, 68–75
    actual parameter, 68
    ambiguous, 41, 72
    argument, 68
    blocking, 128
    evaluation, 70
    statement, 85
    static, 57
    target object, 74
  declaration, 40
  description, 106
  formal parameter, 40
  hiding, 36
    example, 43
    invocation. See method call
    modifier, 42
  name, 40
  nameless, 82
  non-virtual, 42
  overloading, 40
    example, 75
    overriding, 36
    example, 43, 75
  return type, 40
  return value, 40
  sealed, 42
  signature, 40
  static, 42
  virtual, 42
  method-declaration, 40
  method-description, 106
method-modifier, 40
Microsoft C# compiler (example), 3
Min method (Math), 124
MinusOne constant (Decimal), 6
MinValue constant, 6
misplaced semicolon (example), 91
missing value, 114
module, 2
    external, 2
monitor, 120
Monitor class, 122
Mono
    C# compiler (example), 3
    project, 2
Month enum (example), 111
most derived implementation, 74
MoveNext method (IEnumerator<T>), 160
mscorlib assembly, 2
MTAThread (attribute), 178
multi-dimensional array, 28
multiple threads (example), 119, 121
multiplication operator (\*), 57
mutual exclusion (example), 121

\n (new line escape code), 16
n-queens problem (example), 101
name, 4
    of file, 140
    of formal parameter, 40
reserved, 4
Name property
    DirectoryInfo, 140
    FileInfo, 140
named parameter of attribute, 178
nameless method, 82
namespace, 172–173
    declaration, 172
    default, 172
    System, 9, 172
    System.Collections, 158
    System.Collections.Generic, 158, 159
    System.Data.Odbc, 167
    System.Diagnostics, 176
    System.IO, 126, 128
    System.Net, 143
    System.Net.Sockets, 142, 143
    System.Runtime.Serialization, 179
    System.Text, 24, 172
    System.Threading, 118
namespace-declaration, 172
naming convention, 4
NaN (not a number), 10, 124
    constant (Double), 6
natural logarithm, 124
negation operator (-), 57
NegativeInfinity constant (Double), 6
nested
class, 46
    example, 51
type, 32
.netmodule file, 2
network communication, 142
new
    array creation, 26, 57
    class member hiding, 36
    class modifier, 32
    constraint, 148
    delegate creation, 57, 112
    field modifier, 38
    interface member, 106
    interface modifier, 106
    method modifier, 42
    object creation, 57, 64
    struct value creation, 64
new abstract (method modifier), 42
new line escape code (\n), 16
new static (method modifier), 42
new virtual (method modifier), 42
non-static
code, 32
    field. See instance field
method. See instance method
non-virtual method, 42
NonSerialized (attribute), 178
not equal to operator (\!=), 57, 60
    for nullable types, 114
    for strings, 16
NotSupportedException, 116
NotSupportedException, 116, 136, 139, 158, 164
null, 8, 14
null-coalescing operator (??), 57, 114
nullable type, 114–115
Nullable<T> struct, 114, 157
NullReferenceException, 66, 74, 76, 78, 94, 112, 116, 120
number formatting, 22
code, 22
example, 133
NumberFormat property (CultureInfo), 20
numeric type, 6

O notation, 159
object, 38, 54
creation expression, 64
current, 32, 66
initialization, 44
versus struct value (example), 103
object (alias for System.Object), 8
Object class, 8, 36
ObjectDisposedException, 116, 128, 130, 136, 139
Obsolete (attribute), 178
ODBC database connection (example), 167
One constant (Decimal), 6
OpenRead method (FileInfo), 140
OpenText method (FileInfo), 140
OpenWrite method (FileInfo), 140
operator, 56
arithmetic, 58
bitwise, 60
comparison, 60
overloaded, 50
generic (no such thing), 144
in generic type (example), 147
shift, 60
option to compiler, 2
Option<T> struct (example), 157
order of evaluation, 56
out modifier
argument, 68, 70
parameter, 40, 70
output, 126–144
binary, 134
byte-oriented, 136
caracter-oriented, 132
sequential, 132
overflow
checking, 58
floating-point, 58
integer, 58
OverflowException, 6, 10, 26, 58, 98, 99, 116
overloading
of constructors, 44
of indexers, 48
example, 109
of methods, 40
of operators, 50
resolution, 68, 70
compile-time, 72
override (method modifier), 42
overriding a method, 36
example, 35, 43
Pair struct (example), 157
parameter, 14
actual. See argument array, 40, 70
by-value, 40, 70
formal, 40
name, 40
named, of attribute, 178
of indexer, 48
out, 40, 70
passing, 70
example, 71, 73
ref, 40
parameter-constraints, 148
parameter-modifier, 40
parametric polymorphism, 144
params (parameter array), 40, 70
Parent property (DirectoryInfo), 140
Index

arse method (simple types), 6
partial
ordering, 162
type declaration, 174–175
partial type declaration, 174
path (example), 141
PathTooLongException, 116, 140
Pek method, 128
Queue<T>, 170
Stack<T>, 170
TextReader, 130
PekChar method, 128
BinaryReader, 134
percentage formatting, 22
phone prefix codes (example), 87
PI constant (Math), 124
Point class
example, 33, 45
generic (example), 147
polymorphism, parametric, 144
polynomial, generic (example), 151
Pop method (Stack<T>), 170
Position property (Stream), 136
PositiveInfinity constant (Double), 6
postdecrement operator (--), 57, 58
postincrement operator (++), 57, 58
Pow method (Math), 124
pre-processing directive, 176
precedence, 56
predecrement operator (--), 57, 58
preincrement operator (++), 57, 58
prime number server (example), 143
Print method (example), 93
print non-empty strings (example), 93
private member, 34
program, 2
layout, 4
promotion type, 56
property, 48
abstract, 48
access, 57
expression, 76–77
compound assignment, 76
declaration, 48
example, 77
description, 106
generic (no such thing), 144
get-access, 76
set-access, 76
static (example), 77
type, 48
virtual (example), 77
property-description, 106
protected internal member, 34
protected member, 34
public
class, 46
member, 34
Pulse method (Monitor), 122
PulseAll method (Monitor), 122
Push method (Stack<T>), 170
queens problem (example), 101
Queue<T> class, 170
quicksort (example), 113, 153, 155
\r (carriage return escape code), 16
/r (compile option), 2
random access file (example), 137
random numbers
example, 21, 27, 43, 53
Gaussian (example), 125
rank of array, 26, 30
Rank property (Array), 30
RankException, 30, 116
rational number (example), 105
Read method, 128
BinaryReader, 134
Stream, 136
TextReader, 130
read-only
field, 38
example, 27, 51
of struct type, 104
indexer, 48
property, 48
read-write
indexer, 48
property, 48
readability index (example), 19
ReadBlock method (TextReader), 130
ReadBoolean method (BinaryReader), 134
ReadByte method
  BinaryReader, 134
  Stream, 136
ReadBytes method (BinaryReader), 134
ReadChar method (BinaryReader), 134
ReadChars method (BinaryReader), 134
ReadDecimal method (BinaryReader), 134
ReadDouble method (BinaryReader), 134
reading from
  byte array, 139
  byte stream, 130
  file, 130, 138
ReadInt16 method (BinaryReader), 134
ReadInt32 method (BinaryReader), 134
ReadInt64 method (BinaryReader), 134
ReadLine method (TextReader), 130
readonly field modifier, 38
ReadSByte method (BinaryReader), 134
ReadSingle method (BinaryReader), 134
ReadString method (BinaryReader), 134
ReadToEnd method (TextReader), 130
ReadUint16 method (BinaryReader), 134
ReadUint32 method (BinaryReader), 134
ReadUint64 method (BinaryReader), 134
Receive method (Socket), 142
recognizing keywords (example), 169
rectangular array, 28
redundant return statement, 93
ref modifier
  argument, 68, 70
  parameter, 40
reference
  comparison, 8
  to assembly, 2
  type, 8
  variable, 14
/reference (compile option), 2
ReferenceEquals method (Object), 8
reflection, 8, 178
remainder
floating-point, 58, 124
integer, 58
operator (%), 57
RemoteEndPoint property (Socket), 142
remove clause, 52
Remove method
  ICollection<T>, 158
  IDictionary<K,V>, 164
  IList<T>, 164
  String, 18
  StringBuilder, 24
RemoveAt method (IList<T>), 164
RemoveRange method (List<T>), 166
renaming the states of a DFA (example), 169
Replace method (String), 18
replacing character by string (example), 25
representation type of enum type, 110
reserved name, 4
return statement, 92
  redundant, 93
return type
  void, 40
  of delegate, 112
  of method, 40
  of operator, 50
return value, 40
returntype, 40
Reverse method
  Array, 30
    example, 89
  List<T>, 166
RFC1123 date and time format, 22
right alignment (example), 133
right associative, 56, 62
right shift operator (>>), 57, 60
  overloading, 50
Round method (Math), 124
run-time type, 56
Running
  iterator state, 100
  thread state, 118, 119
  running a program, 2
sbyte (simple type), 7, 10
SByte struct type (sbyte), 7
scope, 14
of label, 94
of member, 14, 32
of parameter, 14, 40
of variable, 14
sealed
class, 34
example, 175
method, 36, 42
sealed (class modifier), 34
sealed override (method modifier), 42
search
for empty string (example), 93
for substring (example), 95
in array, 30
SecurityException, 132, 140
Seek method
BinaryWriter, 134
Stream, 136
SeekOrigin enum, 134, 136
segment swapping (example), 31
semicolon, 84
in delegate type declaration, 112
misplaced (example), 91
Send method (Socket), 142
Seq class (example), 47, 51, 101, 109
SeqEnumerator (example), 47
sequence
enumerator (example), 47
iterator (example), 101
of integers (example), 47, 51, 101, 109
sequential
input, 130
output, 132
Serializable (attribute), 178
serialization, 178
example, 179
Serialize method
BinaryFormatter, 128
SoapFormatter, 128
server socket, 142
set intersection closure (example), 171
set-accessor
of indexer, 48
of property, 48
Set<T> class (example), 165
SetLength method
MemoryStream, 139
Stream, 136
SetValue method (Array), 30
shadowing a field, 14
example, 67
shared state, 118
shift operators, 60
short (simple type), 7, 10
short-cut evaluation, 58
Sign method (Math), 124
signature, 40
best, 68
better, 68
of constructor, 44
signed integer type, 44
significant digits of numeric type, 7
simple type, 6
member, 6
variable, 14
Sin method (Math), 124
single quote (’) escape code (\’), 16
Single struct type (float), 7
Sleep method (Thread), 122
Sleeping (thread state), 118, 119
SoapFormatter class, 179
socket, 142
client, 142
communication (example), 143
server, 142
Socket class, 142
SocketException, 142
Sort method
Array, 30
List<T>, 166
sort of array, 30
SortedDictionary<K,V> class, 168
sortedness check (example), 19
source file, 2
example, 175
sparse matrix (example), 49, 79
Index

Split method (String), 18
SPoint struct type (example), 103
SQL query language, 114
Sqrt method (Math), 124
square root (example), 177
stable sort, 30
stack, 54
Stack<T> class, 170
StackOverflowException, 116
StackTrace property (Exception), 116
standard
  conversion, 10–12
  error, 132
  input, 130
  output, 132
Start method (Thread), 122
StartsWith method (String), 18
state, 56, 84
  iterator, 100
  shared, 118
  thread, 118, 119
statement, 84–101
  assignment, 85
  block, 84
  break, 92
  checked, 98
  choice, 86
  continue, 92
  declaration, 84
  do-while, 90
  empty, 84
  expression, 84
  foreach, 88
  goto, 94
  if, 86
  if-else, 86
  labeled, 94
  lock, 120
  loop, 88–91
  method call, 85
  return, 92
  switch, 86
  termination (example), 85
throw, 94
try-catch-finally, 96, 117
unchecked, 98
using, 98
while, 90
yield, 100
STAThread (attribute), 178
static
class, 34
code, 32
constructor, 44
  example, 43
field, 38
  access, 57, 66
  example, 43
  in generic type, 146
member, 32
method, 42
method call, 57
property
  access, 57
  example, 77
static
  class modifier, 34
  field modifier, 38
  method modifier, 42
stream, 126, 136
  byte, 126
  character, 126
  creating, 126
Stream class, 136
StreamReader class, 130
StreamWriter class, 132, 133
string, 16–23
  alignment (example), 21
  builder, 24–25
  character escape code, 16
  comparison, 18, 60
  concatenation, 18, 24, 57
  efficient, 24
  example, 17
  constant, 16
  verbatim, 16
  equality (example), 17
Index

formatting, 20–23
example, 29, 133
indexer, 16

string (alias for System.String), 16
string array file (example), 137
String class, 8, 16
StringBuilder class, 24
example, 25, 51
StringList class (example), 79
StringWriter class, 132

struct

type, 102–106
declaration, 102
default constructor, 102
generic, 156–157
implementing interface, 108
operator overloading, 50
partial declaration, 174
run-time representation, 54
this reference, 104
value, 102
comparison (==, !=), 60
creation expression, 64
parameter passing (example), 73
run-time representation, 54

struct constraint, 148
struct-body, 102
struct-declaration, 102
struct-interface-clause, 108
struct-modifiers, 102
subclass, 36
example, 35
subinterface, 106
Substring method (String), 18
subtraction operator (-), 57
Suspended (iterator state), 100
switch statement, 86
example, 95, 131
synchronization, 120–123
SynchronizationLockException, 116, 120, 122
Synchronized method

textReader, 128
TextWriter, 128
System namespace, 9, 172

System.Collections namespace, 158
System.Collections.Generic namespace, 158, 159
System.Data.Odbc namespace, 167
System.Diagnostics namespace, 176
System.IO namespace, 126, 128
System.Net namespace, 143
System.Net.Sockets namespace, 142, 143
System.Runtime.Serialization namespace, 179
System.Runtime.Serialization.Soap namespace, 179
System.Text namespace, 24, 172
System.Threading namespace, 118
SystemException, 116

\t (tabulator escape code), 16
Tan method (Math), 124
\target (compile option), 2
target of attribute, 178
temperature conversion (example), 133
text

file, writing to (example), 133
stream, 126

TextWriter class, 130
Thermometer class (example), 53

this

constructor call, 44
current object reference, 66
current struct reference, 104
example, 103
in anonymous method, 82
indexer
access, 78
declaration, 48

thread, 118–123
and stack, 54
communication, 118
creation, 118
current, 20
example, 53
safety of input-output, 128
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>118, 119</td>
</tr>
<tr>
<td>transition</td>
<td>118</td>
</tr>
<tr>
<td>Thread class</td>
<td>118, 122</td>
</tr>
<tr>
<td>ThreadInterruptedException</td>
<td>116, 118, 122</td>
</tr>
<tr>
<td>ThreadStart delegate</td>
<td>118</td>
</tr>
<tr>
<td>ThreadStateException</td>
<td>116</td>
</tr>
<tr>
<td>three-valued logic</td>
<td>114</td>
</tr>
<tr>
<td>three-way comparison (example)</td>
<td>63</td>
</tr>
<tr>
<td>throw statement</td>
<td>94</td>
</tr>
<tr>
<td>throwing an exception</td>
<td>94</td>
</tr>
<tr>
<td>time</td>
<td>159</td>
</tr>
<tr>
<td>complexity</td>
<td>159</td>
</tr>
<tr>
<td>constant</td>
<td>159</td>
</tr>
<tr>
<td>formatting as string</td>
<td>22</td>
</tr>
<tr>
<td>linear</td>
<td>159</td>
</tr>
<tr>
<td>logarithmic</td>
<td>159</td>
</tr>
<tr>
<td>quadratic</td>
<td>168</td>
</tr>
<tr>
<td>ToArray method</td>
<td>166</td>
</tr>
<tr>
<td>List&lt;T&gt;, 166</td>
<td></td>
</tr>
<tr>
<td>MemoryStream, 139</td>
<td></td>
</tr>
<tr>
<td>Queue&lt;T&gt;, 170</td>
<td></td>
</tr>
<tr>
<td>Stack&lt;T&gt;, 170</td>
<td></td>
</tr>
<tr>
<td>ToCharArray method (String), 18</td>
<td></td>
</tr>
<tr>
<td>tokenizer (example), 131</td>
<td></td>
</tr>
<tr>
<td>ToLower method (String), 18</td>
<td></td>
</tr>
<tr>
<td>top layers of the type hierarchy, 9</td>
<td></td>
</tr>
<tr>
<td>ToString method</td>
<td></td>
</tr>
<tr>
<td>Enum, 110</td>
<td></td>
</tr>
<tr>
<td>example, 19, 33, 51</td>
<td></td>
</tr>
<tr>
<td>Exception, 116</td>
<td></td>
</tr>
<tr>
<td>Object, 8</td>
<td></td>
</tr>
<tr>
<td>String, 16</td>
<td></td>
</tr>
<tr>
<td>StringBuilder, 24</td>
<td></td>
</tr>
<tr>
<td>ToUpper method (String), 18</td>
<td></td>
</tr>
<tr>
<td>transition of thread state, 118</td>
<td></td>
</tr>
<tr>
<td>traversing</td>
<td></td>
</tr>
<tr>
<td>collection, 161</td>
<td></td>
</tr>
<tr>
<td>dictionary, 161</td>
<td></td>
</tr>
<tr>
<td>Trim method (String), 18</td>
<td></td>
</tr>
<tr>
<td>TrimEnd method (String), 18</td>
<td></td>
</tr>
<tr>
<td>TrimStart method (String), 18</td>
<td></td>
</tr>
<tr>
<td>TrimToSize method</td>
<td></td>
</tr>
<tr>
<td>List&lt;T&gt;, 166</td>
<td></td>
</tr>
<tr>
<td>Queue&lt;T&gt;, 170</td>
<td></td>
</tr>
<tr>
<td>Stack&lt;T&gt;, 170</td>
<td></td>
</tr>
<tr>
<td>try-catch-finally statement, 96, 117</td>
<td></td>
</tr>
<tr>
<td>TryGetValue method</td>
<td></td>
</tr>
<tr>
<td>Dictionary&lt;K,V&gt;, 168</td>
<td></td>
</tr>
<tr>
<td>two’s complement</td>
<td>60</td>
</tr>
<tr>
<td>type, 6–13</td>
<td></td>
</tr>
<tr>
<td>better, 68</td>
<td></td>
</tr>
<tr>
<td>cast</td>
<td></td>
</tr>
<tr>
<td>expression, 57, 80</td>
<td></td>
</tr>
<tr>
<td>illegal, 80</td>
<td></td>
</tr>
<tr>
<td>compile-time, 6, 56</td>
<td></td>
</tr>
<tr>
<td>constructed, 144, 146</td>
<td></td>
</tr>
<tr>
<td>abbreviation for, 154</td>
<td></td>
</tr>
<tr>
<td>declaration</td>
<td></td>
</tr>
<tr>
<td>class, 32</td>
<td></td>
</tr>
<tr>
<td>delegate, 112</td>
<td></td>
</tr>
<tr>
<td>interface, 106</td>
<td></td>
</tr>
<tr>
<td>partial, 174–175</td>
<td></td>
</tr>
<tr>
<td>struct type, 102</td>
<td></td>
</tr>
<tr>
<td>governing, 86</td>
<td></td>
</tr>
<tr>
<td>hierarchy, 8</td>
<td></td>
</tr>
<tr>
<td>top layers, 9</td>
<td></td>
</tr>
<tr>
<td>nested, 32</td>
<td></td>
</tr>
<tr>
<td>nullable, 114–115</td>
<td></td>
</tr>
<tr>
<td>numeric, 6</td>
<td></td>
</tr>
<tr>
<td>of a struct value, 64</td>
<td></td>
</tr>
<tr>
<td>of array, 26</td>
<td></td>
</tr>
<tr>
<td>of enum member, 110</td>
<td></td>
</tr>
<tr>
<td>of expression, 56</td>
<td></td>
</tr>
<tr>
<td>of indexer, 48</td>
<td></td>
</tr>
<tr>
<td>of property, 48</td>
<td></td>
</tr>
<tr>
<td>parameter, 144, 152</td>
<td></td>
</tr>
<tr>
<td>of class, 146</td>
<td></td>
</tr>
<tr>
<td>constraint, 148</td>
<td></td>
</tr>
<tr>
<td>reference, 8</td>
<td></td>
</tr>
<tr>
<td>run-time, 56</td>
<td></td>
</tr>
<tr>
<td>simple, 6</td>
<td></td>
</tr>
<tr>
<td>Type class, 80</td>
<td></td>
</tr>
<tr>
<td>TypeInitializationException, 44, 116</td>
<td></td>
</tr>
<tr>
<td>TypeLoadException, 116</td>
<td></td>
</tr>
<tr>
<td>typeof operator, 57, 80</td>
<td></td>
</tr>
<tr>
<td>\U (Unicode character code), 16</td>
<td></td>
</tr>
</tbody>
</table>
uint (simple type), 7, 10
UInt16 struct type (ushort), 7
UInt32 struct type (uint), 7
UInt64 struct type (ulong), 7
ulong (simple type), 7, 10

Unary operator overloading, 50
UnauthorizedAccessException, 132, 140

Unboxing
- a boxed value, 103
- conversion, 12
- object as value, 104

Unchecked
- expression, 57, 58
- statement, 98

Unchecked context, 58, 98
Unicode character encoding, 16, 182
Universal Character Set, 182
Unlock method (FileStream), 138

Unsigned integer type, 6
User-defined conversion, 52
ushort (simple type), 7, 10

Using
- directive, 154, 172
- example, 107
- statement, 98

UTF-8
- character encoding, 130, 132
- format, 137, 182

\v (vertical tab escape code), 16
Value, 14
- type, 6
- struct type, 102

Value field (KeyValuePair<K,V>), 168
Value-oriented data, 102
Values property (IDictionary<K,V>), 164
ValueType class, 8
- not usable as constraint, 148

Variable, 14
- captured, 82
- declaration, 84
- of reference type, 8, 14
- of simple type, 14
- of struct type, 102

Verbatim string constant, 16
Vertical tab escape code (\v), 16

Vessel class (example), 35

Virtual
- method, 42
- property (example), 77
- method modifier, 42

Void pseudo-type
- as return type, 40
- not in constructed type, 146

Wait method (Monitor), 122
Wait set, 120
Waiting (thread state), 118, 119
- week number (example), 111
- weekday (example), 87, 91, 93, 95, 111
- WeekdayException (example), 117
- where (type parameter constraint), 148
- while statement, 90

White space character, 18
Wordlist (example), 177
Worklist algorithm (example), 171

Write method, 128
- BinaryWriter, 134
- Stream, 136
- TextWriter, 132

Write-only
- indexer, 48
- property, 48

WriteByte method (Stream), 136
WriteLine method, 128
- TextWriter, 132

WriteTo method (MemoryStream), 139
Writing to
- byte array, 139
- file, 138
- text file (example), 133

\x (character code), 16

Xor operator (^), 57

Yield statement, 100
Yield type, 100

Zero constant (Decimal), 6