

Index

COPYRIGHTED MATERIAL

Index

SYMBOLS

- && (double ampersand) expression, 96–97
- || (double pipe) expression, 97
- ! (exclamation point) expression, 97
- ? (question mark) expressions, rules for, 98

A

abbreviations, naming conventions for, 339

abstract accessor

- events, 185
- properties, 183

abstract (keyword), 279

abstract methods, 181

access modifiers, 174

accessibility, 28

accessors, 182–185

additive operators, 25, 114

advantages of C#, 1–2, 5

alias directives, 250

alternative text editors, 15

ambiguities in grammar, 37–38

and operators, 112–119

anonymous method conversions, 109

anonymous methods, 131

Antechinus C# Editor, 15

application startup, 57–58

application termination, 58

argument lists, 124

arithmetic operators, 131–132

array covariance, 205–206

array types, 22–23, 79, 203–204

arrays

- access, 128
- accessing array elements, 205
- creating, 205
- elements, 85–86
- initializers, 206–208
- jagged arrays, 22–23
- members, 62, 205
- overview, 201–203
- rectangular arrays, 22–23
- syntactic grammar, 293–294
- `System.Array` type, 204
- trailing commas, 208

as (keyword), 279

assignment operators, 26, 115, 135

assignments, 198

attributes

- compilation, 237
- Conditional attribute, 238–239
- global attributes, 36
- instances, 236
- named parameters, 232
- naming, 342
- Obsolete attribute, 239–240
- overview, 231–232
- parameter types, 233
- positional parameters, 232
- reserved attributes, 238
- runtime retrieval of attribute instances, 237
- specification, 233–236
- syntactic grammar, 294–296
- usage of, 232–233

B

- base (keyword), 279**
- base types, 120**
- base-access, **128–129**
- base-class **specification, 171**
- binary operator**
 - overload resolution, 118
 - overview, 113, 189
- block **statements, rules for, 91**
- body**
 - interfaces, 212
 - structs, 195
- bool (keyword), 279**
- bool **type, 20, 77**
- Boolean expressions, 138**
- Boolean literals, 47**
- boxing conversion, 80, 102, 198**
- break (keyword), 279**
- break **statement**
 - overview, 156–157
 - rules for, 93
- byte (keyword), 279**
- byte **type, 20, 74–75**

C

- calling generic methods, 251–252**
- Camel case, 338**
- capitalization, 337**
- case (keyword), 279**
- case sensitivity, 338–339**
- cast expressions, 131**
- catch clause, not finding, 229**
- catch (keyword), 279**
- char (keyword), 279**
- char **type, 20, 74–75**
- character literals, 48–49**
- checked (keyword), 279**
- checked **operator, 130**
- checked **statements, rules for, 91**
- circular references, avoiding, 219**
- class (keyword), 279**
- class type, 79**
- class-base **specification, 171**
- class-body, **171**
- classes. See also specific classes**
 - access modifiers, 174
 - attributes, 231–232
 - base-class specification, 171
 - class-base specification, 171
 - class-body, 171
 - class-member-declarations, 172–173
 - constants, 175–176
 - declarations, 169–170, 242–243
 - exception classes, 228
 - finalizers, 191
 - inheritance, 173–174
 - instance constructors, 190–191
 - instance members, 175
 - instance variables in, 86
 - interface types, 171
 - namespaces, organizing classes with, 161–162
 - naming, 341–342
 - new modifier, 174
 - overview, 27–28, 169

- partial declarations, 172
- static constructors, 191
- static members, 174
- structs compared, 196–197
- syntactic grammar, 296–307
- classifications of expressions, 111–112**
- `class-member-declarations`, **172–173**
- closed constructed types, 243**
- closed type, 249**
- CLR (Common Language Runtime), 2, 359–360**
- code blocks, 144**
- comments**
 - delimited comments, 40–42
 - lexical grammar, 276–277
 - nesting comments, 43
 - overview, 40
 - single-line comments, 42–43
 - syntax in XML documentation comments, 363–365
 - tags in XML documentation comments, 365
- Common Type System (CTS), 359**
- compilation**
 - attributes, 237
 - conditional compilation directives, 54
 - conditional compilation symbols, 53
 - unsafe code, 273
- compilation units, 35–37, 162–163**
- compile-time errors, 259**
- conditional AND operator, 25, 114**
- `Conditional attribute`, **238–239**
- conditional attribute class, 239**
- conditional compilation directives, 54**
- conditional compilation symbols, 53**
- conditional logical operators, 133–134**
- conditional methods, 238**
- conditional operator, 25**
- conditional OR operator, 25, 114**
- const (keyword), 279**
- constant expressions, 135–138**
- constants, 28, 175–176**
- constraints, 253–255**
- constructed types**
 - alias directives, 250
 - closed type, 249
 - generics, 249–250
 - members, 250
 - open type, 249
 - overview, 249
 - type arguments, 249
- constructors, 199**
- continue (keyword), 279**
- `continue statement`
 - overview, 157
 - rules for, 93
- conversion operators, 190**
- conversions**
 - anonymous method conversions, 109
 - explicit conversions, 22, 103–107
 - implicit conversions, 22, 99–103
 - method group conversions, 109
 - null type conversions, 109
 - nullable conversions, 109–110
 - overview, 99
 - standard conversions, 107–108
 - user-defined conversions, 108
- cost of using C#**
 - bare minimum to start with C# programming, 7–10
 - free tools, 10–13
 - high-end tools, 15–16
 - .NET Framework, 9–10
 - overview, 7
 - text editor, 8–9
 - UltraEdit, 13–15
 - Windows Notepad, 8–9
- Crimson Editor, 15**
- CTS (Common Type System), 359**
- `Current property`, **262**

D

decimal (keyword), 279

decimal type

- to float/double type conversion, 105
- to integral type conversion, 104
- overview, 20, 77

declaration directives, 54

declaration statements

- local constant declarations, 147
- local variable declarations, 146–147
- overview, 146
- rules for, 91

declarations

- class declarations, 169–170, 242–243
- delegate declarations, 222, 223, 248
- enums, 216–217
- interface declarations, 210–211, 247–248
- local constant declarations, 147
- local variable declarations, 146–147
- namespace declarations, 58–60, 163–164
- overview, 58–60
- struct declarations, 247
- type declarations, 166–167

declared accessibility, 62–63

default constructors, 72

default (keyword), 279

default value expression, 130

default values, 89, 198

defining interfaces, 210

definite assignment

- block statements, rules for, 91
- break statements, rules for, 93
- checked statements, rules for, 91
- continue statements, rules for, 93
- declaration statements, rules for, 91
- do statements, rules for, 92–93
- double ampersand (&&) expressions, rules for, 96–97
- double pipe (| |) expressions, rules for, 97
- exclamation point (!) expressions, rules for, 97
- expression statements, rules for, 91
- foreach statements, rules for, 94

- goto statements, rules for, 93
- if statements, rules for, 91–92
- initially assigned variables, 90
- initially unassigned variables, 90
- lock statements, rules for, 94–95
- overview, 89–90
- question mark (?) expressions, rules for, 98
- return statements, rules for, 93
- rules for determining, 90–98
- simple expressions, rules for, 95–96
- statements, general rules for, 91
- switch statements, rules for, 92
- throw statements, rules for, 93
- try-catch statements, rules for, 93–94
- try-finally statements, rules for, 94
- unchecked statements, rules for, 91
- using statements, rules for, 94
- while statements, rules for, 92
- yield statements, rules for, 98

delegate (keyword), 279

delegate type, 79

delegates

- declarations, 222, 223, 248
- instantiation, 224–225
- invocation list, 223–224
- members, 62
- modifiers, 222–223
- overview, 31, 221–222
- syntactic grammar, 308

delimited comments, 40–42

destructors. See finalizers

diagnostic directives, 54

directives

- conditional compilation directives, 54
- conditional compilation symbols, 53
- declaration directives, 54
- diagnostic directives, 54
- line directives, 55
- overview, 51–53
- pragma directives, 55
- preprocessing expressions, 53

region control directives, 54–55
using, 36

`Dispose` **method**, 262–263

do (keyword), 280

`do` **statement**

overview, 154–155
rules for, 92–93

DotGNU, 3

double ampersand (&&) expression, 96–97

double (keyword), 280

double pipe (| |) expression, 97

double type

to decimal type conversion, 105
to float type conversion, 105
to int type conversion, 105
overview, 20

E

ECMA-334 C# Language Specification, 1, 3

ECMA (European Computer Manufacturers Association), 3

EditPad Lite, 15

EditPad Pro, 15

element access

array access, 128
base-access, 128–129
checked operator, 130
indexer access, 128
new operator, 129
overview, 127
sizeof operator, 129–130
this-access, 128
typeof operator, 129
unchecked operator, 130

else (keyword), 280

empty statements, 145

end point, 142

enterprise tools

overview, 15
Visual C#, 16
Visual Studio, 15–16

entry point, 18, 57

enum (keyword), 280

enumerable interfaces, 259

enumerable objects, 263

enumeration members, 61

enumeration types, 77

enumerator interfaces, 259

enums

circular references, avoiding, 219
declarations, 216–217
members, 218–219
modifiers, 217–218
naming, 342
operators, 220
overview, 31, 215–216
syntactic grammar, 308–309
`System.Enum`, 219
values, 219–220

equality operators, 25, 114, 119

European Computer Manufacturers Association (ECMA), 3

event access, 122–123

event (keyword), 280

events

abstract accessor, 185
field-like events, 185
instance events, 185
interfaces, 213
naming, 343
override accessor, 186
overview, 29, 184–185
sealed accessor, 186
static events, 185
virtual accessor, 185

examples

sample C# code, 3–5
of unsafe code, 269–270

exception classes

exception classes, 228

exceptions

- catch clause, not finding, 229
- handling, 229
- overview, 227
- `System.ArithmeticException`, 228
- `System.ArrayTypeMismatchException`, 228
- `System.DivideByZeroException`, 228
- `System.Exception`, 228
- `System.IndexOutOfRangeException`, 228
- `System.InvalidCastException`, 228
- `System.NullReferenceException`, 228
- `System.OutOfMemoryException`, 228
- `System.OverflowException`, 228
- `System.StackOverflowException`, 228
- `System.TypeInitializationException`, 228
- throwing, 227

exclamation point (!) expression, 97

execution interruptions for `MoveNext` method, 261–262

explicit base interfaces, 211

explicit conversions

- explicit enumeration conversions, 105
- explicit numeric conversions, 103–105
- explicit reference conversions, 106
- explicit type parameter conversions, 107
- overview, 103
- standard explicit conversions, 108
- unboxing conversions, 107
- user-defined explicit conversions, 107

explicit enumeration conversions, 105

explicit interface member implementations, 248

explicit (keyword), 280

explicit numeric conversions

- decimal type to float/double type conversion, 105
- decimal type to integral type conversion, 104
- double type to float type conversion, 105
- float/double type to decimal type conversion, 105
- float/double type to int type conversion, 105
- integral type to integral type conversion, 104
- overview, 103–104

explicit reference conversions, 106

explicit type parameter conversions, 107

expression statements

- overview, 148
- rules for, 91

expressions

- anonymous methods, 131
- arithmetic operators, 131–132
- assignment operators, 135
- Boolean expressions, 138
- cast expressions, 131
- classifications of, 111–112
- conditional logical operators, 133–134
- constant expressions, 135–138
- default value expression, 130
- function members, 121–125
- logical operators, 133–134
- member lookup, 119–120
- null coalescing operator, 134
- and operators, 112–119
- overview, 24–26, 111
- primary expressions, 125–130
- relational/type testing operators, 132–133
- results of, 112
- shift operators, 132
- syntactic grammar, 309–317
- unary expressions, 131
- values for, 112

extern (keyword), 280

extern-alias-directive, 164–165

F

false (keyword), 280

field initialization, 178, 199

field-like events, 185

fields

- field initialization, 178
- instance fields, 177
- overview, 24, 28, 176–177
- `readonly` fields, 177
- static fields, 177
- variable initialization, 178
- volatile fields, 177–178

finalizers

- overview, 29, 191
- structs, 199

finally (keyword), 280**fixed (keyword), 280****fixed modifier, 270–272****float (keyword), 280****float type**

- to decimal type conversion, 105
- to int type conversion, 105
- overview, 20

floating-point types, 76–77**for (keyword), 280****for statement, 155–156****foreach (keyword), 280****foreach statement**

- overview, 156
- rules for, 94

format standardization, 3**free tools, 10–13****function members**

- argument lists, 124
- event access, 122–123
- indexer access, 123
- instance constructor invocation, 123
- method invocation, 121
- operator invocation, 123
- overload resolution, 125
- overview, 121
- property access, 122

G**generic methods**

- calling, 251–252
- inference of type arguments, 251–252
- overview, 250–251
- signatures, 251
- virtual generic methods, 251

generics

- advantages of, 242
- class declarations, 242–243

class members, 245

constraints, 253–255

constructed types, 249–250

delegate declarations, 248

explicit interface member implementations, 248

instance type, 244–245

interface declarations, 247–248

operators in generic classes, 247

overloading in generic classes, 246–247

overview, 31–32, 241

protected members, access to, 246

static constructors, 246

static fields, 246

struct declarations, 247

syntactic grammar, 317–318

templates in C++ compared, 241

type parameters, 243–244

where not to use, 252–253

GetEnumerator method, 263–264**global attributes, 36****goto (keyword), 280****goto statement**

- overview, 157
- rules for, 93

grammar

ambiguities, 37–38

lexical grammar, 37, 39–40, 275–292

syntactic grammar, 37, 292–331

unsafe code, extensions for, 332–335

H**Hajlsberg, Anders (C# principal designer), 2****“Hello, World!” program, 3–5, 17–18****high-end tools, 15–16****history of C#, 2****I****IDE (Integrated Development Environment), 35****identifiers, 44–45, 277–279****identity conversions, 100**

if (keyword), 280

if statement

- overview, 148–149
- rules for, 91–92

implementation-defined behavior, 360–361

implementations in development for C#, 3

implicit constant expression conversions, 103

implicit conversions

- boxing conversions, 102
- identity conversions, 100
- implicit constant expression conversions, 103
- implicit enumeration conversions, 101
- implicit numeric conversions, 100
- implicit reference conversions, 101–102
- implicit type parameter conversions, 102–103
- overview, 99–100
- standard implicit conversions, 107–108
- user defined implicit conversions, 103
- user-defined implicit conversions, 108

implicit enumeration conversions, 101

implicit (keyword), 280

implicit numeric conversions, 100

implicit reference conversions, 101–102

implicit type parameter conversions, 102–103

in (keyword), 280

index signatures, 63

indexer access, 123, 128

indexers

- overloading, 64
- overview, 29, 186–187

inference of type arguments, 251–252

inheritance

- overview, 30, 173–174
- structs, 197

initializers, 206–208

initially assigned variables, 90

initially unassigned variables, 90

instance constructors

- invocation, 123
- overloading, 64
- overview, 29, 190–191
- signatures, 63

instance events, 185

instance fields, 177

instance members, 175

instance methods, 180

instance properties, 182

instance type, 244–245

instance variables

- in classes, 86
- overview, 86
- in structs, 87

instantiation, 224–225

int (keyword), 280

int type, 20, 74–75

integer literals, 47

integral types

- byte type, 74–75
- char type, 74–75
- int type, 74–75
- to integral type conversion, 104
- long type, 74–75
- overview, 74–76
- sbyte type, 74–75
- short type, 74–75
- uint type, 74–75
- ulong type, 74–75
- ushort type, 74–75

Integrated Development Environment (IDE), 35

interface (keyword), 280

interface types, 171

interfaces

- body, 212
- declarations, 210–211, 247–248
- defining, 210
- events, 213
- explicit base interfaces, 211
- members, 62, 212
- methods, 212
- modifiers, 211
- naming, 342
- overview, 30–31, 209
- properties, 212–213
- structs, 195
- syntactic grammar, 318–320

internal (keyword), 280

interoperability with legacy code, 359–360

invocation expressions, 127

invocation list, 223–224

is (keyword), 280

ISO/IEC 23270 standard, 3

iteration statements

- do statement, 154–155
- for statement, 155–156
- foreach statement, 156
- overview, 154
- while statement, 154

iterator block, 258–259

iterators

- compile-time errors, 259
- Current property, 262
- Dispose method, 262–263
- enumerable interfaces, 259
- enumerable objects, 263
- enumerator interfaces, 259
- execution interruptions for MoveNext method, 261–262
- GetEnumerator method, 263–264
- iterator block, 258–259
- MoveNext method, 260–262
- overview, 32, 257–258
- this, 260
- yield break statement, 262
- yield return statement, 261
- yield type, 260

J

jagged arrays, 22–23

jump statements

- break statement, 156–157
- continue statement, 157
- goto statement, 157
- overview, 156
- return statement, 158
- throw statement, 158

K

keywords

- abstract, 279
- as, 279
- to avoid, 339–341
- base, 279
- bool, 279
- break, 279
- byte, 279
- case, 279
- catch, 279
- char, 279
- checked, 279
- class, 279
- const, 279
- continue, 279
- decimal, 279
- default, 279
- delegate, 279
- do, 280
- double, 280
- else, 280
- enum, 280
- event, 280
- explicit, 280
- extern, 280
- false, 280
- finally, 280
- fixed, 280
- float, 280
- for, 280
- foreach, 280
- goto, 280
- if, 280
- implicit, 280
- in, 280
- int, 280
- interface, 280
- internal, 280
- is, 280
- lexical grammar, 279–282
- lock, 280

keywords (continued)

- long, 280
- namespace, 280
- new, 280
- null, 281
- object, 281
- operator, 281
- out, 281
- override, 281
- overview, 46
- params, 281
- private, 281
- protected, 281
- public, 281
- readonly, 281
- ref, 281
- return, 281
- sbyte, 281
- sealed, 281
- short, 281
- sizeof, 281
- stackalloc, 281
- static, 281
- string, 281
- struct, 281
- switch, 281
- this, 281
- throw, 281
- true, 281
- try, 281
- typeof, 282
- uint, 282
- ulong, 282
- unchecked, 282
- unsafe, 282
- ushort, 282
- using, 282
- virtual, 282
- void, 282
- volatile, 282
- while, 282

L

labeled statements, 145–146

language structure

- attributes, 36
- comments, 40–43
- compilation units, 35–37
- directives, 36, 51–55
- global attributes, 36
- grammar, 37–40
- line terminators, 40
- namespace member declarations, 36
- source files, 35–37
- tokens, 36, 43–51
- whitespace, 43

learning C#, 5

lexical grammar

- comments, 276–277
- identifiers, 277–279
- line terminators, 282
- literals, 282–286
- operators/punctuators, 286–288
- overview, 36, 39–40, 275–276
- pre-processing directives, 288–292
- unicode escape characters, 292
- white space, 292

lexical grammar (keyword), 279–282

lifted operators

- equality operators, 119
- overview, 118
- relational operators, 119
- unary operators, 118

line directives, 55

line terminators, 40, 282

literals

- Boolean literals, 47
- character literals, 48–49
- integer literals, 47
- lexical grammar, 282–286
- null literal, 49
- overview, 46

real literals, 48
string literals, 49

local constant declarations, 147
local variable declarations, 146–147
local variables, 88–89
lock (keyword), 280
lock **statements, rules for, 94–95**
logical AND operator, 25, 114
logical operators, 133–134
logical OR operator, 25, 114
logical XOR operator, 25, 114
long (keyword), 280
long **type, 21, 74–75**

M

member access, 62, 126–127
member lookup
base types, 120
overview, 119–120
members
array members, 62, 205
class members, 61–62
constructed types, 250
delegate members, 62
enumeration members, 61, 218–219
interface members, 62, 212
namespace members, 61
overview, 60–61, 172–173, 245
struct members, 61, 195–196
memory management, 66–67
method group conversions, 109
methods
abstract methods, 181
body, 181
instance methods, 180
interfaces, 212
invocation, 121
method body, 181

naming, 343
overloading, 64
override methods, 180–181
overview, 28, 178–179
parameters, 179–180
sealed methods, 181
signatures, 63–64
static methods, 180
virtual methods, 180
Microsoft Intermediate Language (MSIL), 36, 359–360
modifiers
delegates, 222–223
enums, 217–218
interfaces, 211
overview, 170–171
structs, 195
Mono, 3
MoveNext **method, 260–262**
multiplicative operators, 25, 114

N

the name C#, explanation of, 1
named parameters, 232
namespace (keyword), 280
namespace-member-declaration, **36, 166**
namespaces
compilation units, 162–163
declarations, 58–60, 163–164
extern-alias-directive, 164–165
members, 61
namespace-member-declaration, 166
naming, 341
organizing classes with, 161–162
overview, 66, 161
qualified-alias-member, 167–168
scope controlled with, 162
type declarations, 166–167
using directives, 165–166

naming conventions

- abbreviations, 339
- attribute naming, 342
- Camel case, 338
- capitalization, 337
- case sensitivity, 338–339
- class naming, 341–342
- enumeration type naming, 342
- event naming, 343
- interface naming, 342
- keywords to avoid, 339–341
- method naming, 343
- namespace naming, 341
- overview, 337
- parameter naming, 343
- Pascal case, 337
- property naming, 343
- static field naming, 342
- uppercase, 338

nesting comments, 43

.NET, 2–3

.NET Framework, 9–10

new (keyword), 280

new **modifier**, 174

new **operator**, 129

Notepad ++, 15

NoteTab, 15

null coalescing operators, 114, 134

null (keyword), 281

null literal, 49

null **type**, 79

null type conversions, 109

nullable conversions, 109–110

nullable types, 32, 80–81

O

object (keyword), 281

object **type**, 21, 79

objects, how types are treated as, 70

Obsolete **attribute**, 239–240

open constructed type, 243, 249

operator (keyword), 281

operators. See also unary operators

- additive operators, 25, 114
- assignment operators, 115
- binary operator, 113, 189
- binary operator overload resolution, 118
- conditional AND operator, 25, 114
- conditional operator, 25
- conditional OR operator, 25, 114
- conversion operators, 190
- enums, 220
- equality operators, 25, 114, 119
- and expressions, 112–119
- in generic classes, 247
- invocation, 123
- lifted operators, 118–119
- list of, 286–288
- logical AND operator, 25, 114
- logical OR operator, 25, 114
- logical XOR operator, 25, 114
- multiplicative operators, 25, 114
- null coalescing operators, 114
- overloading, 64, 115–118
- overview, 29, 112–113, 187–189
- pointers, 268–269
- precedence, 113–115
- primary operators, 25, 114
- relational operators, 119
- relational/type-testing operators, 25, 114
- shift operators, 25, 114
- signatures, 64
- ternary operator, 113
- tokens, 50–51
- types of, 113

out (keyword), 281

out **parameter**, 70

output parameters, 24, 88, 180

overload resolution, 125

overloading

- in generic classes, 246–247
- indexers, 64
- instance constructors, 64

methods, 64
 operators, 64, 115–118
 overview, 64
override accessor
 events, 186
 properties, 183–184
override (keyword), 281
override methods, 180–181
overview of C#
 accessibility, 28
 advantages of C#, 1–2, 5
 classes, 27–28
 constants, 28
 delegates, 31
 enums, 31
 events, 29
 expressions, 24–26
 fields, 24, 28
 finalizers, 29
 generics, 31–32
 “Hello, World!” program, 17–18
 history of C#, 2
 implementations in development for C#, 3
 indexers, 29
 inheritance, 30
 instance constructors, 29
 interfaces, 30–31
 iterators, 32
 learning C#, 5
 methods, 28
 nullable types, 32
 operators, 29
 parameters, 24
 properties, 28–29
 source code, examining, 18
 standardization, 3
 statements, 26–27
 static classes, 30
 static constructors, 30
 structs, 30
 types, 19–23
 variables, 23–24

P

parameters
 arrays, 24, 180
 attribute parameters, 233
 for methods, 179–180
 naming, 343
 output parameters, 24
 overview, 24
 reference parameters, 24
 value parameters, 24
params (keyword), 281
parenthesized expressions, 126
partial declarations, 172
Pascal case, 337
platform/OS portability, 359
pointers, 268–269
portability
 implementation-defined behavior, 360–361
 interoperability with legacy code, 359–360
 overview, 359, 362
 platform/OS portability, 359
 simplified deployment, 359
 undefined behavior, 360
 unspecified behavior, 361–362
positional parameters, 232
pragma directives, 55
precedence, 24–26, 113–115
predefined reference types, 19, 21
predefined types, 19–21
predefined value types, 20–21
pre-processing directives, 288–292
preprocessing expressions, 53
primary expressions
 element access, 127–130
 invocation expressions, 127
 literals, 125
 member access, 126–127
 overview, 125
 parenthesized expressions, 126
 simple names, 126
primary operators, 25, 114
private (keyword), 281

Programmer's Notepad, 15

properties

- abstract accessor, 183
- access, 122
- accessors, 182–184
- instance properties, 182
- interfaces, 212–213
- naming, 343
- override accessor, 183–184
- overview, 28–29, 181–182
- sealed accessor, 184
- static properties, 182
- virtual accessor, 183

protected (keyword), 281

protected members, access to, 246

public (keyword), 281

punctuators

- list of, 286–288
- overview, 50–51

Q

qualified-alias-member, **167–168**

question mark (?) expressions, rules for, **98**

R

reachability, **142–144**

readonly fields, **177**

readonly (keyword), **281**

real literals, **48**

rectangular arrays, **22–23**

ref (keyword), **281**

ref parameter, **70**

reference parameters, **24, 87–88, 180**

reference types

- array type, 79
- boxing, 80
- class type, 79
- delegate type, 79
- null type, 79
- object type, 79

overview, 78

string type, 79

unboxing, 80

value types compared, 69–70

region control directives, 54–55

relational operators, 119

relational/type-testing operators, 25, 114, 132–133

reserved attributes, 238

return (keyword), 281

return statement

overview, 158

rules for, 93

runtime retrieval of attribute instances, 237

S

sbyte (keyword), 281

sbyte type, **21, 74–75**

scope

controlled with namespaces, 162

overview, 64–66

sealed accessor

events, 186

properties, 184

sealed (keyword), 281

sealed methods, 181

selection statements

if statement, 148–149

overview, 148

switch statement, 149–153

shift operators, 25, 114, 132

short (keyword), 281

short type, **21, 74–75**

signatures

generic methods, 251

index signatures, 63

instance constructor signatures, 63

method signatures, 63–64

operator signatures, 64

and overloading, 64

overview, 63

simple expressions, rules for, 95–96

simple names, 126

simple types, 73

simplified deployment, 359

single-line comments, 42–43

sizeof (keyword), 281

`sizeof` operator, 129–130, 272

source code, examining, 18

source files, 35–37

specification, 233–236

`stackalloc`, 273

stackalloc (keyword), 281

standard conversions, 107–108

standard explicit conversions, 108

standard implicit conversions, 107–108

standard library, 345–357

standardization, 3

statement lists, 144–145

statements

code blocks, 144

declaration statements, 146–147

empty statements, 145

end point, 142

expression statements, 148

general rules for, 91

iteration statements, 154–156

jump statements, 156–158

labeled statements, 145–146

overview, 26–27, 139–141

reachability, 142–144

selection statements, 148–153

statement lists, 144–145

syntactic grammar, 320–327

types of, 141

using statement, 158–159

yield statement, 159–160

static classes, 30

static constructors

generics, 246

overview, 30, 191

structs, 199

static events, 185

static fields

naming, 342

overview, 177, 246

static (keyword), 281

static members, 174

static methods, 180

static properties, 182

static variables, 85

string (keyword), 281

string literals, 49

`string` type, 21, 79

struct declarations, 247

struct (keyword), 281

struct members, 61

struct types, 72–73

`struct-declaration`, 194–195

structs

assignments, 198

body, 195

boxing, 198

class compared, 196–197

constructors, 199

default values, 198

field initializers, 199

finalizers, 199

inheritance, 197

instance variables in, 87

interfaces, 195

members, 195–196

modifiers, 195

overview, 30, 193–194

static constructors, 199

`struct-declaration`, 194–195

syntactic grammar, 327–329

`this` variable, 198

unboxing, 198

when to use, 199–200

switch (keyword), 281

switch statement

overview, 149–153

rules for, 92

syntactic grammar

- arrays, 293–294
- attributes, 294–296
- classes, 296–307
- delegates, 308
- enums, 308–309
- expressions, 309–317
- generics, 317–318
- interfaces, 318–320
- overview, 37, 292–293
- statements, 320–327
- structs, 327–329
- types, 329–331
- variables, 331

syntax in XML documentation comments, 363–365

- `System.ArithmeticException`, **228**
- `System.Array` **type**, **204**
- `System.ArrayTypeMismatchException`, **228**
- `System.DivideByZeroException`, **228**
- `System.Enum`, **219**
- `System.Exception`, **228**
- `System.IndexOutOfRangeException`, **228**
- `System.InvalidCastException`, **228**
- `System.NullReferenceException`, **228**
- `System.OutOfMemoryException`, **228**
- `System.OverflowException`, **228**
- `System.StackOverflowException`, **228**
- `System.TypeInitializationException`, **228**
- `System.ValueType` **class**, **71–72**

T

tags in XML documentation comments, 365

templates in C++ compared to generics, 241

ternary operator, 113

text editor, 8–9

this (keyword), 281

this **variable**, **198, 260**

this-access, **128**

throw (keyword), 281

throw **statement**

- overview, 158
- rules for, 93

throwing exceptions, 227

tokens

- identifiers, 44–45
- keywords, 46
- literals, 46–49
- operators, 50–51
- overview, 43
- punctuators, 50–51
- Unicode escape sequences, 43–44

trailing commas, 208

transformation, 36

true (keyword), 281

try (keyword), 281

try-catch **statements, rules for**, **93–94**

try-finally **statements, rules for**, **94**

typeof (keyword), 282

typeof **operator**, **129**

types

- arguments, 249
- array types, 22–23, 79
- bool type, 20, 77
- boxing, 80
- byte type, 20, 74–75
- char type, 20, 74–75
- class type, 79
- conversions, 22
- decimal type, 20
- declarations, 166–167
- double type, 20
- float type, 20
- instance type, 244–245
- int type, 20, 74–75
- long type, 21, 74–75
- names, 66
- object type, 21
- objects, how types are treated as, 70
- out parameter, 70
- overloading, 22
- overview, 19, 19–23, 74–76
- parameters, 243–244
- predefined reference types, 19, 21
- predefined types, 19–21
- predefined value types, 20–21

ref parameter, 70
 reference types, 19, 69–70, 78–80
 sbyte type, 21, 74–75
 short type, 21, 74–75
 string type, 21
 syntactic grammar, 329–331
 types of, 69
 uint type, 21, 74–75
 ulong type, 21, 74–75
 unboxing, 80
 ushort type, 21, 74–75
 using, 76
 value types, 19, 69–77, 80–81

U

uint (keyword), 282
 uint **type, 21, 74–75**
ulong (keyword), 282
 ulong **type, 21, 74–75**
UltraEdit, 13–15
unary expressions, 131
unary operators
 lifted operators, 118
 overload resolution, 117
 overview, 25, 113, 189
 precedence, 114
unboxing conversions, 80, 107, 198
unchecked (keyword), 282
 unchecked **operator, 130**
 unchecked **statements, rules for, 91**
undefined behavior, 360
unicode escape characters, 43–44, 292
unsafe code
 advantages of, 266
 compilation, 273
 disadvantages of, 266
 example, 269–270
 extensions for, 332–335
 fixed modifier, 270–272
 overview, 265
 pointers, 268–269

 sizeof operator, 272
 stackalloc, 273
 unsafe contexts, 266–268
unsafe contexts, 266–268
unsafe (keyword), 282
unspecified behavior, 361–362
unwrapping, 109
uppercase, 338
user-defined conversions, 103, 107, 108
user-defined explicit conversions, 107
user-defined implicit conversions, 103, 108
ushort (keyword), 282
 ushort **type, 21, 74–75**
 using **directives**
 overview, 165
 using-alias-directive, 165
 using-namespace-directive, 166
using (keyword), 282
 using **statement**
 overview, 158–159
 rules for, 94
 using-alias-directive, **165**
 using-namespace-directive, **166**

V

value parameters, 24, 87, 180
value types
 bool type, 77
 boxing, 80
 decimal type, 77
 default constructors, 72
 enumeration types, 77
 floating-point types, 76–77
 integral types, 74–76
 nullable types, 80–81
 overview, 70–71
 reference types compared, 69–70
 simple types, 73
 struct types, 72–73
 System.ValueType class, 71–72
 unboxing, 80

variable initialization, 178

variables

- array elements, 85–86
- categories of, 84–85
- default values, 89
- definite assignment, 89–98
- instance variables, 86–87
- local variables, 88–89
- output parameters, 88
- overview, 23–24, 83–84
- reference parameters, 87–88
- static variables, 85
- syntactic grammar, 331
- value parameter, 87

virtual accessor

- events, 185
- properties, 183

virtual generic methods, 251

virtual (keyword), 282

virtual methods, 180

Visual Studio, 15–16

void (keyword), 282

void pointers, 268

volatile (keyword), 282

volatile fields, 177–178

W

while (keyword), 282

while statement

- overview, 154
- rules for, 92

whitespace, 43, 292

Windows Notepad

- overview, 8–9
- writing code in, 10–13

wrapping, 109

writing code

- with free tools, 10–13
- in Windows Notepad, 10–13

X

XML documentation comments

- overview, 363
- syntax, 363–365
- tags, 365

Y

yield break statement, 262

yield return statement, 261

yield statement

- overview, 159–160
- rules for, 98

yield type, 260

