Index

All bold page numbers indicate definitions

A

aBackspaceChar variable, 74
AbeLincoln file, 592
abstract classes, **461**–464, 478
  abstract methods, 461, 468
  data or methods for derived classes, 469
  nonabstract methods, 461, 468
abstract keyword, 461
abstract methods, **398,** 413
  abstract classes, 461, 468
  interfaces, 466–469
  overriding, 461
AcceptButton property, 106
accessibility, **273**–274, 302
access keys, **566,** 585, 599
access modifiers
  classes, 356
  properties, 365
accessors, **364,** 413
AccessSomeNames program, 680–683
access specifiers, **111,** 130, 357
ACCOUNTING constant, 166–167
AccountNumber class, 471
AccountNumber objects, 471
AccountNum property, 528
accumulated totals, **210**–211, 220
ActiveLinkColor property, 555
actual parameters, **282,** 303, 312
add and assign shortcut operator (+=), **65,** 91, 211, 615, 624, 629
AddBooks program, 395
AddCheckDigit() method, 471
AddDays() method, 578
addition operator (+), 63, 392, 615
add keyword, 15
AddMonths() method, 578
AddRangeObjectsDemo program, 573
AddYears() method, 578
AdvertisingMessage() method, 378
ageOverMinimum variable, 170
age variable, 158–160
alert escape sequence (\a), 74
aliases, **48,** 50, 90, 450
  memory, 314
  variables, 314–315
alias keyword, 15
AllowDrop property, 544
Alt property, 640
ambiguous methods, **333**–334, 345
American Standard Code for Information Interchange (ASCII), **709,** 716
AMethod() method, 334, 342–343
ancestors, **428,** 477
Anchor property, 544, 582
AND operator (&&), **154**–156, 171, 173, 180, 237
  combining with OR operator (||), 158–162
anExceptionInstance object, 499
Animal class, 428, 462–463, 466–467
animal class, 354
Animal parameter, 464
anInt variable, 59
annualIncome variable, 69
annualSalary variable, 57
anotherValue variable, 65
answer variable, 506
anUnsignedInt variable, 59
AnyLegalClassName identifier, 16
Append file mode, 666
applicable methods, **329,** 345
application classes, **13,** 40
ApplicationException class, 492, 525
applications
  See also programs
  classes, 354
  runnable, **13**
application software, **2,** 38
area field, 375
Area property, 376
args variable, 297
ArgumentException class, 491
arguments, **11,** 39, 246–247, 270, 281–284
  constructors, 458–459
  data types, 283–284
  default, 285
  method call, 276
  named, **338**–342
  passing by reference, 312
  passing by value, 312
  positional, **338**
  strings, 11
  unnamed, 337–338
arithmetic
  array elements, 233
  numeric type conversion, 71–73
  parentheses ( ) in expressions, 64
ArithmeticException class, 491, 494
arithmetic operators, **63**–64, 91, 392
  shortcut arithmetic operators, 64–66
ArrayDemo1 program, 236
ArrayDemo2 program, 250–252
array elements, **228,** 260
array of references, 397
arrays, 29, **228,** 260, 675
  accessing elements, 232–235
  arithmetic on elements, 233
  automatically altering value, 234
  bool fields, 231
  character fields, 231
  comparing value to valid values in, 237
  creation, 235–236
  cycling through elements, 234–235
  declaring, 228–232
  default values, 231

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- arrays (*continued*)
  - descending order elements, 398
  - duplicate values, 247
  - GUI programs, 258–259
  - identifiers, 229
  - increasing value of elements, 232
  - index, **228**
  - initializer list, **231**
  - initializing, 230–232
  - iteration variable, **234**
  - jagged, **256**–257
  - length, 233–234, 245
  - matching values in, 398
  - methods for manipulating, 245–249
  - multidimensional, **252**–256
  - naming, 229
  - navigating, 234–235
  - nondefault values for elements, 231
  - numbering elements, 229, 230
  - numeric fields, 231
  - one-dimensional, **252**
  - parallel arrays, **238**
  - passed by reference, **294**
  - passing to methods, 293–296
  - range match, 247
  - rectangular, **253**
  - reserving memory locations for, 229
  - resizing, 229, 234
  - reversing order of items, 249–250
  - searching with loops, 237–245
  - separating fields into strings, 672
  - serializable, 684
  - single-dimensional, **252**
  - size, 231
  - sorting, 246–248
  - subscripts, **228,** 230, 675
  - three-dimensional, 256
  - two-dimensional, **252**
  - usage, 235–236
  - values for elements, 228–232
  - variables, 228
  - zeroth element, 230
- arrays of objects, 397–405
  - searching, 398–402
  - sorting, 398–402
- Array.Sort() method, 403
- ArrayTypeMismatchException class, 491
- Artist class, 461
- ASCII (American Standard Code for Information Interchange), **709**
  - decimal and binary values for characters, 710–712
- assemblers, 3
- assemblies, **356,** 412
- assembly language, 3
- assignment, **50,** 90
- assignment operator (=), **50,** 90, 147–148, 171, 359–360, 615
  - right-associative, **705**
- associativity, **64,** 91, **705**–706
- aTabChar variable, 74
- at runtime. *See* runtime
- attributes, **5,** 7, 39, 355
  - associating information with code, 684
- audio clips, 104
- auto-implemented property, **368**–369, 373, 413
- automatically generated code, 547–550
- automatic properties, **368**–369, 413
- Automobile class, 7–9
- a variable, 278
- aWorker object, 381, 389

B

- BackColorChanged event, 632
- BackColor property, 105–106, 109, 128, 544, 560, 565, 570, 572, 585, 589, 592, 594, 646
- BackgroundImage property, 544
- backing field, **365,** 413
- backslash escape sequence (\\), 74
- backspace escape sequence (\b), 74
- *Bad command or file name* error message, 23
- BaileysForm form, 561
- balanceDue variable, 65
- Balance property, 527–528
- BankAccountBalanceAdjusted() method, 629–630
- BankAccount class, 8, 527–528, 628–630
- BankAccount object, 629–630
- BankBalance form, 218
- bankBalance variable, 72, 196
- bankBal variable, 65, 203, 219
- base class, **427,** 474–475, 477
  - abstract classes, 461
  - constructors, 456–460
  - derived classes, 443–448
  - implicit conversion, **448**–449
  - overriding members, 439–444
  - size, 427
- base keyword, 458
- base 2 numbers, 708
- base 10 numbers, 707
- BASE_PRICE constant, 567
- base 16 shorthand notation, **74**–75, 92
- base 16 system, **713,** 716
- BASIC
  - array elements, 230
  - interpreters, 3
- BedAndBreakfast form, 593–96
- BedAndBreakfast program, 561–565
  - check boxes, 587–594
- bedrooms variable, 254, 256
- Begin member, 678
- behaviors, **5,** 7, 39
- bellAireCheckBox object, 587
- BellAire form, 591
- belleAireCheckBox_CheckedChanged() method, 591
- BelleAire CheckBox object, 588
- belleAireDescriptionLabel label, 589
- BelleAireForm form, 588–589
- belleAirePriceLabel label, 589
- BelleAire Suite check box, 591, 597
- betterness rules, **329,** 345
- bigFont object, 553
- biggerNum variable, 168
- BillPatients() method, 431
- binary files, **656,** 668, 697
- binary form, 392
- BinaryFormatter class, 684
- BinaryFormatter objects, 683–687
- sbinary numbering system, **708**–709, 716
  - base 2 numbers, 708
- binary operators, **63,** 91, 168
  - left-associative, **705**
  - overloaded, 392–393
- BinaryReader class, 668
- binary search, **246**–247, 261
- BinarySearchDemo program, 246
- BinarySearch() method, 246–247, 250, 270
  - array of objects, 398–402
- BinaryWriter class, 668
- bin folder, 35
- bits (binary digits), **708,** 714, 716
- bitwise operators, **158,** 180
- bitwise OR operator (|), 553
- black box, **8,** 39, **271,** 302
- block comments (/* */), **18,** 40
- blocks, **144,** 180
  - aligning braces ({ }), 145
  - exiting immediately, 241
  - if-else statements, 149
  - K & R style, 145
  - multiple statements, 144
  - single statement, 145
  - unreachable, **510**
- blueButton button, 645
- blueToolStripMenuItem_Click() method, 585–586
- BMethod() method, 343
- Bold property, 553
- bonus parameter, 341

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- bonus variable, 157
- Book class, 377–379, 393–396
- Book objects, 377–378, 396
- Book parameter, 396
- Book return type, 394
- Boole, George, 68
- bool data type, 48–49, 68–69, 91, 142, 288–289
- Boolean expressions, 141–142
  - combining AND operator (&&) and OR operator (||), 158–162
  - compound, 154–156
  - negating, 169–170
  - side effects, **157**–158
- Boolean fields, 381
- Boolean logical AND operator (&), **157**–158, 180
- Boolean logical inclusive OR operator (|), **157**–158, 180
- boolean.Parse() method, 86
- Boolean values, 140–141, 288–289
- Boolean variables, **68,** 70, 91
  - comparison operators, **68**–69
  - naming, 68
- BorderDemo1 program, 324–325
- BorderDemo2 program, 327–328
- Bottom property, 545
- Bounds property, 545
- Box class, 390
- BreakfastOptionForm class, 595
- BreakfastOptionForm form, 594–596
- BreakfastOptionForm_Load() method, 595
- break keyword, 164–165
- break statements, 165, 241, 515
- BUDGET2016 class, 16
- Budget2016 class, 15
- BudgetForm form, 106
- bugs, **3,** 38
- Build, Build Solution command, 27, 34
- Build failed error message, 120
- building variable, 256
- Build Solution (Ctrl + Shift + B) keyboard shortcut, 27
- built-in methods, 289
  - output parameter, 317–320
  - overloaded, 330
- BusDriver class, 468
- BusinessOffice class, 359
- ButtonBase class, 565
- button1 button, 108–110, 409–410, 547, 719
- button2 button, 642–643
- button2_Click() event, 643
- button1_Click() event handler, 643
- button1_Click() method, 111–112, 114, 118, 274
- button2_Click() method, 111
- Button() constructor, 549
- Button control, 107–108, 114, 551
  - focus, 116
  - properties, 126
- button1 object, 111
- Button objects, 407–409, 474, 542–543, 549, 582, 608, 632–634, 642–643, 645, 688, 691, 693–694
- Button property, 638
- buttons, 107, 124–129, 176, 608–609
  - adding functionality, 109–112
  - Click() event, **611,** 642
  - clicking, 110–111
  - default event, 110
  - double-clicking, 634
  - focus, 645
  - functionality, 127–128
  - method executing when pressed, 109
  - naming, 108
  - positioning, 126
  - properties, 109
  - renaming, 111
  - resizing, 126, 594, 688, 691
  - summing numbers, 114
  - text, 547
- b variable, 278
- byte data type, **56**–57, 59, 71, 90
  - automatic conversions, 283
- bytes, **657,** 697, **709,** 714, 716

C

- C++, 5
  - compared to C# programming language, 10
  - declaring array variable, 228
  - function pointers, 613
- CalcArea() method, 376
- CalcPay() method, 286–288
- CalcPay program, 287
- CalcPhoneCallPrice() method, 290–292
- calculatePayroll() method, 611
- CalculateWithholdingTax() method, 5
- CalendarForeColor property, 580
- CalendarMonthBackground property, 580
- calendar object, 578
- called method, **270,** 302
- calling, **5,** 12, 38, **270,** 302
- calling method, **270,** 302
- callLength variable, 292
- call stack, **521,** 533
  - listing methods in, 521–525
  - tracing Exception objects through, 521–525
- camel casing, **3,** 38
- CancelButton property, 106
- CanFocus property, 545
- CanRead property, 665
- CanSeek property, 665
- CanSelect property, 545
- CanWrite property, 665
- Capture property, 545
- Car class, 448
- CarLoan class, 432–433, 446–447, 460
- CarLoan() constructor, 460
- CarLoan objects, 432, 446
- Carpet class, 374, 376
- Carpet objects, 375
- carriage return escape sequence (\r), 74
- case keyword, 164
- case labels, **164**–165, 180
- case sensitivity, 3
- case structure. *See* switch structure
- catch block, **498**–502, 505–512, 514–516, 519, 533, 668
  - break statement, 515
  - displaying information about exception, 503
  - e.Message, 524
  - Exception type, 499
  - multiple, 507, 509
  - rethrowing exceptions, **529**–531
  - return statement, 515
  - unreachable, **510**
  - WriteLine() method, 524
  - XxxException placeholder, 499
- catch clause, 498
- catching exceptions, 498–503, 505–506
  - multiple exceptions, 507–511
- catch keyword, 498
- catch() method, 499
- Cat class, 462–463, 466–467
- C# Code Editor, 716–718
  - code snippets, **719**
  - readability aids, 720
  - wavy underlines, 719
- C# compiler, 24–25
- cd (change directory) command, 31, 35
- changeButton button, 608, 611
- changeButton_Click() method, 609, 611–612
- Changed event, 623–624
- ChangedEventHandler delegate, 622–623
- ChangedEventHandler event handler, 622
- Changed field, 624

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Changed() method, 623
- changeLabel() method, 611
- changeOutputButton button, 128–129
- changeOutputButton_Click() method, 128
- character fields, initial values, 381
- characters, 56, 73–75, **662,** 698
  - array with subscripts, 232
  - ASCII (American Standard Code for Information Interchange), **709,** 710–712
  - assigning array to string, 231
  - base 16 shorthand notation, **74**–75
  - bits (binary digits), **708**–709
  - case sensitivity, 196
  - character set, **662**
  - char variables, 74
  - EBCDIC (Extended Binary Coded Decimal Interchange Code), **709**
  - hexadecimal shorthand notation, **74**–75
  - nonprinting, **74**
  - numbers as, 73
  - sorting, 398
  - strings, 74
  - Unicode, 74–75, **709**
- character set, **662,** 698
- char data type, 48, **56,** 71, 73–75, 90, 92, 671
  - automatic conversions, 284
- char variables, 74
- CheckBox control, 107
- check boxes, **565**–568, 587–594, 599
- checkBox1 object, 566
- checkBox2 object, 566
- CheckBox objects, 543, 565–568, 587–594
- check digits, 471
- CheckedChanged() method, 565, 591, 596
- checked list boxes, 572–576
- CheckedListBox objects, 543, 572–576
- Checked property, 565–568
- Child class, 468
- child class, **427,** 477
  - accessing parent class data, 435
  - inheriting from parent class, 430–431, 451
- CIS constant, 166–167
- CivilWarSoldier class, 469
- class access modifier, **356,** 412
- class body, 356–357
- class client, **355,** 412
- class definition, **356,** 412
- class diagrams, 426
- classes, **7,** 12, 39, 354
  - abstract classes, **461**–464
  - access modifiers, 356
  - application classes, **13**
  - application programs, 354
  - class body, 356–357
  - class definition, **356**
  - class headers, 14, **356**
  - concrete class, **461**
  - constructors, 361
  - conventional names, 15
  - creation, 356–358, 370–373
  - curly braces ({}), 14, 21
  - declaring, 356
  - default constructor, **381,** 390, 456
  - destructors, **405**–407
  - enumerations, 81
  - extending, 429–431, 433–434
  - fields that are objects of other classes, 359
  - fragile, **438**
  - grouping, 12
  - help, 122
  - identifiers, 14–16, 356
  - illegal names, 16
  - implementing interfaces, 399, 468
  - inheritance, **8, 424**–426, 461
  - instances, **7,** 39, **354,** 405–407
  - instance variables, 356–358
  - instantiating objects, 354, 424
  - instantiation, **354**
  - methods, **270,** 276–277, 357–358
  - named behaviors for, 399
  - naming, 8, 14
  - non-application classes, **13**
  - objects, 377
  - overloaded constructors, 387–388
  - overloaded operators, 393
  - parameterless constructor, 382–383
  - private data fields, 357–358
  - properties, 39
  - repeatedly using from same namespace, 19–20
  - return type, 286
  - sealed classes, **470**
  - sorting methods, 271
  - static classes, 475
  - string holding name of, 452
  - unconventional names, 16
- class headers, 14, **356,** 412
- class keyword, 16, 356
- class user, **355,** 412
- clerk object, 431, 434
- Click delegate, 636
- Click event, 632–634, 638, 642–643, 646
- click event, **111,** 130, **611,** 648
- clickLocationLabel label, 638
- Click() method, 112, 122, 175–176, 270, 274, 299–300, 569, 585, 596, 633–634, 646
- ClickParameterInfo program, 299–300
- Clicks property, 638
- clients, **271,** 302
- Close() method, 693
- closing files, **663,** 676, 698
- Closing() method, 339–340
- COBOL, referring to array elements, 230
- code
  - attributes associating information with, 684
  - automatically generated, 547–550
  - catch block, **498**–499
  - correcting errors, 120–121
  - expanding or condensing, 104
  - line numbers, 121
  - locating line of, 548
  - reusable, 426
  - try block, **498**–499
- code bloat, **272,** 302
- CodeDomSerializerException class, 490
- code refactoring, **118,** 131
- code snippets, **719,** 720
- code window, 109
- color
  - background, 565
  - forms, 558–560
  - GUI applications, 560
  - source code, 720
- Color class, 558–560
- ComboBox class, 576
- combo boxes, 572–**576,** 599
- ComboBox objects, 543, 572–576
- command line, **23,** 29–30, 40
  - compiling and executing programs, 31–32
  - incorrect subdirectory or folder, 24
  - namespaces, 28
  - program comments, 36
  - running programs, 25
  - using statements, 28
- command prompt, **23**–25, 40
- commenting out statements, **17,** 40
- comments, 40, 549
  - block comments (/* */), **18,** 40
  - line comments (//), **18,** 40
  - programs, 36
  - XML-documentation format contents (< >), **18**
- CommissionEmployee class, 425–427, 429, 435–438, 443, 449, 456–458

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- CommissionEmployee() constructor, 458
- CommissionEmployee objects, 430, 435
- commissionRate field, 425, 429, 435
- CommissionRate property, 435, 436
- commissionRate variable, 171–172
- Common Language Runtime exception classes, 492
- ComparableEmployeeArray program, 401–402
- Compare() method, 77–79
- CompareThreeNumbers project, 152–153
- CompareTo() method, 78–79, 398–402, 404
- comparison operators, **68**–69, 92
- compiler errors, 490
- compilers, **3,** 38
  - just in time (JIT), **22**
- compiling programs, 22–30
  - command prompt, 23–25
  - high-level programming languages, 3
  - IDE (Integrated Development Environment), 25–28
  - intermediate language (IL), 22
- completion mode, **717**–718, 720
- Component class, 474, 542–543
- composed delegates, 612, **615**–621, 648
- composition, **359,** 412
- compound Boolean expressions, 154–156
  - short-circuit evaluation, **156**–157
- compound expressions and if statements, 154–160
- ComputeGross() method, 341–342
- ComputePaycheck() method, 281
- computer files, **656,** 697
- computers
  - hardware, **2**
  - measuring storage, 714–716
- computer simulations, **6,** 39
- ComputeTotalPay() method, 341–342
- concatenating strings, **53,** 55, 90, 392
- concrete class, **461,** 478
- conditional AND operator (&&), **154**–156, 180
- conditional operator (?:), **168**–169, 180
  - right-associative, **705**
- conditional OR operator (||), **156**–157, 180
- Connelly, Marc, 542
- Consolas, 117
- console applications, 122–123
- Console class, 12, 19, 274, 474, 664
- Console.Error class, 664
- Console.In class, 664–665
- console input, 83–84
- Console.Out class, 664–665
- constants, **48,** 90
  - assigning value to, 81, 166
  - character values, 73
  - enumerations, **81**–83
  - floating-point constants, 553
  - floating-point numbers, 60
  - int data type, 57
  - int data type and implicit conversion, 72
  - methods, 278, 300
  - naming, 81
  - public access modifiers, 375
  - scope, **278**
  - subscripts, 233
  - unnamed, 48
- const field, 386
- const keyword, 80, 375
- const modifier, 386
- constructor initializer, **385,** 413
- constructors, **380**–386, 413
  - automatically generated, 361, 458
  - base class, 456–460
  - constructor initializer, **385**
  - controls, 549
  - default constructor, 361, **381**
  - derived class, 456–460
  - initial field values, 381
  - initializing variables, 457
  - objects, 456
  - overloading, 382–384
  - parameterless constructor, **382**
  - passing parameters to, 381–382
  - readonly modifier, 386
  - requiring arguments, 458–459
  - throwing exceptions, 528
  - user-created, 381
- ContainerControl class, 474
- ContainerControl objects, 583–584
- Container property, 545
- ContainsFocus property, 545
- contBreakfastButton RadioButton, 594, 596
- contextual keywords, **365,** 413
- ControlAdded event, 632
- Control class, 474, 542–543, 572
- Control objects, 542–543, 554
- Control property, 640
- ControlRemoved event, 632
- controls, **102,** 130, 542–546, 587, 599
  - aligning, 581
  - anchoring, 582, 584
  - check boxes, **565**–568, 587–594
  - checked list boxes, 572–576
  - combo boxes, 572–576
  - constructors, 549
  - dates and time, 577–580
  - default event, **554**–555
  - default font, 117
  - default names, 118
  - deleting, 108
  - docking, 584
  - double-clicking, 555
  - dragging onto forms, 108, 112, 547, 550
  - event handling, 632–636
  - event sender, **610**
  - focus, **116,** 549, **642**
  - fonts, 551–554
  - group boxes, 583–**584**
  - handling multiple events, 643–644
  - layout on form, 581–584
  - link label, **554**–557
  - list boxes, 572, **573**–576
  - listing, 107
  - locating, 581
  - managing multiple, 642–644
  - menu strips, 585–586
  - monthly calendar, 577–580
  - mouse events, 637–639
  - multiple, 114
  - naming, 118–119
  - panels, 583–**584**
  - picture boxes, **569**–571
  - predefined colors, 558–560
  - preventing conflicts, 549
  - properties, 611
  - protected properties, 544
  - public properties, 544–546
  - radio buttons, **565**–568, 594–596
  - relocating, 108
  - renaming, 118–119
  - snap lines, **581**
  - tab order, **116**
  - universal properties, 474
- control statement, **144,** 180
- ConversionWithParse program, 318
- ConversionWithTryParse program, 319
- Convert class, 84–86, 317
- ConvertEmployeeToString() method, 452
- Convert.ToInt32() method, 85, 288, 330, 517
- CountContributions program, 258–259
- counted loops, **195,** 220
- counter variable, 64–65
- Courier New, 117
- C programming language, 4
- C# programming language, 5, **9**–11, 39
  - case sensitivity, 3
- CreateDirectory() method, 660
- CreateEmployee class, 361

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

CreateEmployee program, 361–362
CreateEmployee2 program, 366
Create file mode, 666
CreateFileName program, 674–675
Create() method, 658
CreateNew file mode, 666
CreateSomeEmployees program, 383–384
CreateStudents2 class, 388
CreateStudents.cs file, 372–373
CreateStudents3 program, 403–405
CreateText() method, 658
CreateTwoBooks program, 379
credits field, 439–441
Credits property, 440–441
csc (C Sharp compiler) command, 23–24, 35
cs file extension, 22, 24, 656
CSV (comma-separated values) files, **669,** 698
culture, **62,** 91
CultureInfoClass class, 62
Currency format specifier, 61
currency values, 62
Current member, 678
CursorChanged event, 632
Cursor property, 545
CustomFormat property, 580
custStatus variable, 69

D

DailySpecial program, 177–178
data, 657
  characters, **662**
  data hierarchy, **662**
  fields, **663**
  files, 663
  formatting in GUI applications, 116–117
  organization, 662–663
  records, **663**
  sequential access files, 668–671
data files, **656,** 697
data hierarchy, **662,** 698
Data.txt file, 657
data types, **48,** 90
  arguments, 283–284
  betterness rules for conversion, 329
  bool data type, 68–69
  enumerations, 81–83
  explicitly overriding, **72**
  floating-point data types, 59–62
  fully qualified name, 50
  implicit numeric conversions, 71–73
  integral data types, **56**–57
  intrinsic types, **48**
  largest and smallest values, 48–49
  method names, 325
  numeric type conversion, 71–73
  objects, 359–360
  one-dimensional arrays, 253
  overloading operators, 392
  parameters, 281
  properties, 365
  return type, 286
  serializable, 684
  simple types, **48**
  strings, 85–86
  two-dimensional arrays, 253
  type precedence, **71**
  type-safe, **83**
  unifying type, **71**
  variables, 50, 57
DateChanged() method, 577–578
Date class, 359
Date fields, 359
dates, 577–580, 658
DateTime objects, 578, 658
DateTimePicker objects, 577–580, 658
dayBox text box, 177
DayOfWeek enumeration, 81–82
dead code, **510,** 533
Debug, Start Without Debugging command, 27, 28, 34
Debug folder, 35
debugging programs, **3,** 38
debugging tools in IDE (Integrated Development Environment), 25
Deciduous class, 428
decimal data type, 48, **60,** 71, 91
Decimal format specifier, 61
decimal numbering system, **707,** 716
  base 10 numbers, 707
decision making
  assignment operator (=), 171
  common errors, 171–174
  dual-alternative decisions, **149**–151
  equal to operator (==), 171
  event-driven programs, 176
  flowcharts, 139
  GUI applications, 175–176
  if-else statements, **149**–151
  if statements, **141**–147
  NOT operator (!), 173–174
  AND operator (&&), 173
  OR operator (||), 173
  range checks, **171**–172
  switch structure, **163**–167
decision structure, **140,** 180
  control statement, **144**
decrementing, **194,** 220
decrement operator (--), 91
DeductTax() method, 275
DeductUnionDues() method, 275
default arguments, 285
default case, 165
default constructor, 361, **381,** 390, 397, 413, 456
default event, **554**–555, 599
default font, 117
default keyword, 164
DEFAULT_VALUE constant, 512
default value of objects, **381,** 413
definite loops, **195,** 199, 220
del delegate, 612
del1 delegate, 615
del2 delegate, 615
del3 delegate, 615
delegate keyword, 613
delegate objects, 621
delegates, 611, **612,** 648
  assigning to another delegate, 615
  combining, 615
  creation, 616–618
  declaring, 613
  encapsulating instance methods, 628–629
  encapsulating methods, 613
  passing reference to methods, 612, 614
  removing delegate from, 615
  user-declared events, 622–627
Delete() method, 658, 660
DELIM constant, 672
delimiters, **669,** 671, 688, 698
DeliverBaby() method, 431
deliverRadioButton_CheckedChanged() method, 567–568
deluxeBreakfastButton RadioButton, 594, 596
DemoAnimals program, 464
DemoAnimals2 program, 464
DemoBankAccountEvent class, 630
DemoBankEvent program, 628–629
DemoCarLoan program, 433–434
DemoCarLoan2 program, 445–448
DemoCarLoan3 program, 459–460
DemoEmployeeDestructor program, 406
DemoEmployees class, 430
DemoEscapeSequences program, 76
DemoExtensionMethod program, 471–472

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

DemoLoan program, 432–433
DemoLogo program, 279–280
DemoObjectInitializer2 program, 391
DemoORAndAND file, 161–162
DemoProjectInitializer program, 389
de Morgan, Augustus, 170
DemoSalesperson program, 436–438
DemoSalesperson2 program, 444
DemoSalesperson3 program, 449
DemoSalesperson4 program, 456–457
DemoStudentEvent2 class, 626–627
DemoStudentEvent program, 625
DemoStudents program, 442
DemoVariables class, 58
DemoVariables.cs file, 58–59
DemoVariables2 program, 67–68
DemoVariables3 program, 70–71
DemoVariables project, 58–59
DemoWorking program, 467–468
denom variable, 508
department variable, 173
Depth property, 390
derived class, **427,** 461, 474–475, 477
  abstract methods, 461
  accessing base class methods and properties, 443–444
  constructors, 456–460
  extending, 427–428
  implicit conversion to base class, **448**–449
  inheritance, 435
  overriding methods, 469
  overriding virtual methods or property, 441–442
  size, 427
  using base class members, 445–448
  visible methods or properties, **442**
deserialization, **684**–687, 698
Deserialize() method, 685
Designer.cs file, 104–105, 611
Designer screen, 591
design time, **115,** 130
destructors, **405**–407, 413
DetermineTaxRate() method, 522
Developer Command Prompt command, 23
dinnerPrice variable, 208–209
dir (directory) command, 31, 35
directories, **657,** 660–661, 698
Directory class, 657, 660–661
DirectoryInformation program, 660–661
discountButton_Click() method, 175
Discount class, 617
DiscountDelegateDemo program, 616–618
DiscountDelegateDemo2 program, 619–621
DiscountDelegate object, 617, 619
discountRangeLowLimits array, 244–245
discounts array, 243
DismissedEmployee class, 426
DisplayArgs program, 297
DisplayCompanyLogo() method, 279–280
DisplayEmployeeData() method, 340, 362–363
DisplayEmployeeData program, 362–363
DisplayGreeting() method, 449
Display() method, 372–373, 388, 403, 443
DisplayNetPay() method, 275
DisplayObjectMessage() method, 450
displayOutputButton button, 126, 129
displayOutputButton_Click() method, 128
DisplayParameter() method, 313
DisplayPrice() method, 518
DisplayReferenceParameter() method, 315–316
DisplaySalesTax() method, 281–285
DisplaySize() method, 336
DisplaySomeMoney program, 52
DisplayStrings() method, 320–321
DisplayWithBorder() method, 325–326
DisplayWithBorderUsingInt() method, 325
Dispose() method, 690
Disposing property, 545
DiverseObjects program, 450–451
divide and assign shortcut operator (/=), 65
DivideByZeroException class, 494, 496, 501–503, 507–511
DivideByZeroException object, 493–494
DivisionBasedOnMajor class, 167
division by zero, 495–498
division operator (/), 63
dockable or floating windows, 107
Dock property, 545, 582
Doctor class, 431
documenting programs, 17–18
doesEmployeeHaveDependents variable, 148
Dog class, 7–9, 427, 448, 462–463, 466–467
do keyword, 202
do loops, **202**–204, 210, 216, 220
DomesticPet class, 428
double array, 229, 252
Double class, 86, 354
double data type, 48, **59**–60, 71, 91, **229,** 287–289, 328–329, 333–334, 448, 669, 672
double quotation mark escape sequence (\"), 74
double variable, 288
do...while loop, **202,** 220
  See also do loop
DragDrop event, 632
DragEnter event, 632
DragLeave event, 632
DragOver event, 632
dual-alternative decisions, **149**–151, 180

E

EARLIEST_YEAR constant, 446
EBCDIC (Extended Binary Coded Decimal Interchange Code), **709**
Ecma International web site, 10
elements, **228,** 260
  accessing, 232–235
  assigning values, 228–232
  cycling through, 234–235
  increasing value, 232
  numbering, 229–230
  passing single to method, 293–294
  performing arithmetic on, 233
else-if statements, 172
e.Message, 524
empArray array, 397
empID parameter, 385
Employee class, 15, 356–361, 363, 366–369, 380–381, 383, 385–386, 388, 397, 400, 402, 406, 424–427, 429, 431, 434–436, 443, 449–454, 456–458, 465–467, 663, 669, 684–685, 687
Employee() constructor, 361, 381, 383, 397
EmployeeData.txt file, 672
Employee() method, 380
Employee objects, 358–359, 362, 364, 381, 397, 399–401, 406, 425, 430, 434, 452–453, 455, 668, 686
Employee parameter, 363, 449
empty body, **194,** 220
Empty field, 622
EnabledChanged event, 632
Enabled property, 109, 129, 545
encapsulating methods, 612
encapsulation, **8,** 39, 612
END constant, 670–371, 681
End member, 678

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Enter event, 632
EnterInvoices program, 688–691
entryID variable, 247
entryString variable, 318, 320
enumerations, **81**–83, 92
  assigning values to constants or variables, 82
  data types, 81–83
  GUI applications, 177–178
  identifiers, 81
  switch structures, 166–167
  valid values, 82
enum keyword, 81
Environment.Exit() method, 515
e object, 610–611
Equals() method, 77, 79, 451
  overloading, 453
  overriding, 453–455
equal to operator (==), 69, 77, 147–148, 171
equivalency, testing for, 147–148
Error list, 103
error list, 120
error messages and IDE (Integrated Development Environment), 25
escape characters, 117
escape sequences, **74,** 76, 92
EventArgs argument, 610, 622, 629, 638
EventArgs class, 622
EventArgs.Empty instance, 623
EventArgs object, 629
EventArgs parameter, 623, 625, 637
EventDemo program, 609–612
event driven, **608**
event-driven programs, 648
  decision making, 176
  loops, 217
  runtime, **176**
EventHandler delegate, 525–527
event handlers, **609**–612, 648
  associating with multiple controls, 643–644
  event receiver, **610**
  names, 609
  user-declared, 622–627
EventHandler statements, 636
event handling, 608–612
  controls, 632–636
  handling multiple events, 643–644
  keyboard events, 639–641
  mouse events, 637–639
event-handling methods, 121–122, 300
EventListener class, 624–627, 629–631
EventListener object, 630
event listeners, 629–631
event receiver, **610,** 648
events, **111,** 130, **608,** 648
  associating method with multiple, 646–647
  clicking objects and generating, 555
  connecting to actions, 611–612
  evoking, 648
  firing, **608,** 648
  generalizing method name, 643
  invoking, **623**
  listing, 634–635
  raising, **608,** 648
  triggering, **608,** 648
  user-declared, 622–627
event sender, **610,** 648
event wiring, **611**–612, 648
Evergreen class, 427–428
evoking events, 648
exabytes, 715
Exception class, 490, 492, 494, 499, 502–503
  extending, 525–527
  subclass messages, 526
  user-created, 525–528
exception handling, **490**–494, 505–506, 532, 668
  object-oriented, 498–501
  thrown from outside methods, 517–520
  traditional *versus* object-oriented, 497–498
Exception objects, 494, 497–498, 500–501, 505, 514–515, 517, 525, 526
  information about, 503, 521
  tracing through call stack, 521–525
exceptions, **490**–494, 532
  catching, 498–503, 505–506
  catching multiple, 507–511
  finally block, **514**–516
  multiple, 510
  propagating, **521**
  purposely causing, 495–497
  rethrowing, **529**–531
  throwing, 498–503, 505–506, 519, 528–529
  unhandled, 318, 515–516
ExceptionsOnPurpose.exe file, 496
ExceptionsOnPurpose program, 495–497
ExceptionsOnPurpose2 program, 505–506
ExceptionsOnPurpose3 program, 511–512
Exception types, 499, 502, 507, 509–510, 525, 528
executing programs, 22–30
execution-time errors, 490
.exe files, 25, 656
Exists() method, 658, 660
explicit cast, **72,** 92
explicitly overriding data types, **72,** 92
exposing, **665,** 698
Extended Binary Coded Decimal Interchange Code (EBCDIC), **709,** 716
extended class, **427,** 477
eXtensible Markup Language. *See* XML (eXtensible Markup Language)
extension methods, **470**–472, 478

F

fault tolerant, **494,** 533
favoriteButton button, 645
fields, **355,** 412, **663,** 698
  abstract classes, 461
  access specifiers, 357
  converting to text, 683
  delimiters, **669**
  inheritance, 428
  initial values, 381
  private access, 367, 373–374
  protected access, **436**–438
  public access, 374–376
  retrieving values, 364
  separating into array of strings, 672
  setting, 364
  static, 375
fields array, 672
File, Exit command, 35
File, Save All command, 34
FileAccess enumeration, 666
FileAccess.Read member, 671
FileAccess.ReadWrite member, 671
File class, 657–659
FileInfo class, 658
filenames, 24, 688
File objects, 682
file position pointer, **677,** 698
files, **656,** 663, 697
  accessing information about, 658–659
  binary files, **656**
  closing, 515–516, **663,** 676, 698
  creation, 674–675
  creation time, 657
  CSV (comma-separated values) files, **669**
  data files, **656**
  data organization, 662–663
  defining characteristics, 669

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

directories, **657**
file position pointer, **677**
folders, **657**
key field, **663**
last-modified time, 657
listing, 104–105
names, 657
opening, **663,** 698
path, **657**
permanent storage devices, **656**
priming read, **670**
random access files, **663**
reading from, **657,** 676–677, 697
record delimiter, 669
sequential access files, **663**
size, 657
sorting methods, 271
storing, 657
text files, **656**
text input and output, 665
writing to, **657,** 698

FileStatistics program, 659

FileStream class, 665–666, 685, 689

FileStream file, 666

FileStream objects, 666, 668–669, 671, 674–676, 692

finally block, **514**–516, 533

FindEmployees program, 678–679

FindPriceWithForLoop program, 239

FindPriceWithWhileLoop program, 242

fingerprints, 455

firing events, **608,** 648

FirstClass class, 12–16

FirstClass console application, 11–13

firstDel delegate, 613, 615, 617, 619

firstName variable, 76

fixed-pitch fonts, **117,** 131

Fixed-point format specifier, 61

float argument, 552

Float class, 354

float data type, **59,** 71, 91, 284, 552

floating-point constants, 553

floating-point data types, 59–62
decimal data type, **60**
double data type, **59**
float data type, **59**

floating-point numbers, **59**–62, 90
constants, 60
dividing by zero, 492
formatting, 60–62
significant digits, **59**
standard numeric format strings, **60**–61

floor variable, 254, 256

flowcharts, **138,** 179
decision making, 139
decision structure, **140**
parallelograms, 142
sequence structure, **138**

focus, **116,** 130, 549, **642,** 648

Focused property, 545

folders, **657,** 660–661, 698

FontChanged event, 632

Font class, 551–554

Font() constructor, 552

Font dialog box, 563

Font objects, 553–554

Font property, 105, 109, 117, 128, 544–545, 547, 549, 551, 563, 570, 585, 590, 592, 594, 633

fonts
bold, 553
changing, 117
controls, 551–554
default, 117
fixed-pitch, **117,** 131
GUI applications, 560
labels, 590
Microsoft Sans Serif, 552, 563
monospaced, **117,** 131
names, 552
proportional, **117,** 131
resizing, 563
selecting, 551
sizes, 552
styles, 553
underlined, 553

FontStyle enumeration, 81, 553

Font window, 551

foreach statements, 234–235

ForeColorChanged event, 632

ForeColor property, 105–106, 109, 128, 544–545, 560

for keyword, 199

for loops, **199**–201, 216, 220, 660
comparing value to valid values in array, 237
displaying integers 1 through 10, 199
exiting early, 241
improving efficiency, 240–241
loop body, 199, 200
loop control variable, 199, 214
navigating arrays, 234–235
nested, 204, 206
order of execution, 208
searching arrays, 238–240

formal parameters, **282,** 303, 312

FormatException class, 491, 496, 511, 517

FormatException object, 513

Format property, 580

format specifiers, **61,** 91

format strings, **53**–54, 90, 116
alignment and field size, 55
displaying number as, 62
length, 56

Form class, 105, 216–218, 258, 473–474

Form1 class, 410, 473, 646

Form1.cs[Design] tab, 129, 565, 591, 593, 646

Form.cs file, 638

Form1.cs file, 104, 593, 596, 640, 646

Form1.cs node, 104–105

Form Designer, 103, 109, 114
dragging controls, 547
form name, 105
unwanted event-handling methods, 121

Form1.Designer.cs file, 104, 409, 547, 564, 693

form feed escape sequence (\f), 74

Form1 file, 104

Form1 form, 105, 547, 639, 691

Form1_KeyUp() method, 640

Form objects, 407–408, 473, 542, 560–565, 569–570, 574, 577, 585, 589, 591, 608, 633, 640, 688, 691
layout, 581–584
properties, 550

forms, **102,** 130
background color, 565
buttons, 107–112, 124–129, 126, 176, 608–609
check boxes, **565**–568, 587–594
checked list boxes, 572–576
color, 558–560
combo boxes, 572–576
controls, **102,** 542–546, 633
creation, 102–106
date and time, 577–580
default event, 555
default names, 118
deleting controls, 108
dragging controls onto, 108, 112, 547, 550
failing to close before reexecuting program, 121
labels, **112**–115, 124–129, 560–565, 589, 640
layout, 581–584
list boxes, 572, **573**–576
maximum and minimum size, 583
menu strips, 585–586
modifying, 105
monthly calendar, 577–580
multiple controls, 114

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

forms (*continued*)
  naming, 102, 105, 118–119, 125
  nonstatic methods, 300
  picture boxes, **569**–571
  properties, 105, 125
  radio buttons, **565**–568, 594–596
  reading data from text files, 691–696
  relocating controls, 108
  renaming, 106
  resizing, 633, 645
  saving, 126
  strings, 106
  text, 551–554
  text boxes, **113**–115, 176
FormWithALabelAndAButton program, 547–550
forward declarations, 10
fragile, **438,** 477
Frame property, 555
fullBreakfastButton RadioButton, 594, 596
fully qualified name, **274,** 303
function pointers, 613
functions, 5

G

gallonsOfGas variable, 499–500, 502
garbage, **211,** 220
General format specifier, 61
get accessors, **364**–368, 373, 413, 429, 446, 452
GetCreationTime() method, 658, 660
GetData() method, 403–404
GetDirectories() method, 660
GetEmployeeIndentification() method, 452
GetFiles() method, 660
GetGreeting() method, 429, 443–444
GetHashCode() method, 451, 454–455
get keyword, 15
GetLastAccessTime() method, 658, 660
GetLastWriteTime() method, 658, 660
GetLength() method, 296
GetPrice() method, 288
GetSelected() method, 575
getter, **365,** 413
GetType() method, 451–452, 454, 503
get variable, 365
gigabytes, **657,** 697, 715–716
global keyword, 15
Goodbye() method, 613, 615
GotFocus event, 632
governing type, **164,** 180
gpa field, 623
Gpa property, 624
gradePointAverage field, 371
GradePointAverage property, 372–373
graphical control elements, **542,** 599
graphical user interfaces. *See* GUIs (graphical user interfaces)
graphics, displaying, 569–571
Greater than operator (>), 69
Greater than or equal to operator (>=), 69
Greeting class, 613
Greeting2 class, 615
GreetingDelegate() delegate, 613
Greeting program, 614
Greeting2 program, 615–616
greetingsButton button, 216
greetingsButton_Click() method, 216–217
greeting variable, 444
GreetMethod() method, 613
gross parameter, 341–342
grossPay variable, 287
gross variable, 287
GroupBox class, 584
group boxes, 583–**584,** 599
Grow() method, 428
GUI applications
  arrays, 258–259
  automatically generated methods, 299–300
  closing, 115
  colors, 560
  controls, 542–546
  correcting errors, 120–121
  deciding to use, 122–123
  decision making, 175–176
  enumeration, 177–178
  event driven, **608**
  event handling, 632–636
  event-handling methods, 122
  executing, 115
  failing to close form before reexecuting, 121
  fonts, 560
  formatting data, 116–117
  if-else statements, 175
  if statements, 175
  inheritance, 473–475
  keyboard events, 639–641
  logical errors, 121
  loops, 216–219
  Main() method, 110–111
  memory, 122
  methods, 299–300
  mouse event handling, 637–639
  objects, 407–410
  scope, 300
  switch structures, 175, 177–178
  text file creation, 688–691
GUI objects, 542
  events, **111**
GUIs (graphical user interfaces), **6,** 39
  objects, 6, 102
  switching between visual environment and code, 109

H

hardware, **2,** 38
has-a relationship, **359,** 412
HasChildren property, 545
hash codes, **455,** 478
HeartSurgeon class, 431
heatingBill variable, 48
Height property, 106, 390, 544–545
Hello class, 21–22
HelloClass program, 270, 277
Hello.cs program, 21–22, 31, 35
Hello.exe file, 31–32, 35
HelloForm form, 125
helloLabel label, 127–128, 586, 608
Hello() method, 613, 615
Hello namespace, 33
Hello project, 32
HelloVisualWorld project, 124–129
Help documentation, 122
HelpRequested event, 632
hexadecimal, 92
Hexadecimal format specifier, 61
hexadecimal numbering system, **713**–714, 716
  base 16 system, **713**
hexadecimal shorthand notation, **74**–75
hidden or visible windows, 107
hide, **441,** 477
HIGH constant, 147
high-level programming languages, **2**–3, 38
horizontal tab escape sequence (\t), 74
hot keys, 566
hours parameter, 341
hoursWorked variable, 4, 69

I

IAttackable interface, 468–469
IComparable interface, 399–400, 404
IDE (Integrated Development Environment), 22–**23,** 29–30, 40
  automatically generated code, 25, 29, 547–550, 564–565

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

changing properties of multiple objects, 553
colors of code, 25, 30
compiling and executing program, 32–36
compiling code, 25–28
correcting errors, 30
debugging tools, 25
double-clicking controls, 633
Error list, 103
error messages, 25, 27
executing program, 115
Form Designer, 103
forms, 102–106, 124–129
information about methods, 330
menu bar, **25,** 103
menu options, 27
naming project, 26, 103
New Project window, 103
Properties window, 103
shortcuts, 27
Solution Explorer, 103
syntax errors, 27
text editors and, 28–29
Toolbox tab, 103
using statements, 29
View menu, 103–104
identifiers, **4,** 28, 38
arrays, 229
classes, 356
enumerations, 81
methods, 275–276
names, 14–16
objects, 359–360
parameters, 281
@ prefix, 14
properties, 365
selecting, 14–16
text editors, 28
variables, 50
verbatim identifiers, **14**
idNumber field, 357–358, 361–367, 371, 386, 400–402, 404, 406
IdNumber property, 364–369, 372–373, 385, 388–389, 404
idNumbers array, 246–247
idNum field, 424–425, 429, 434–435, 623
IdNum property, 434, 452–454, 624
if-else statements, **149**–153, 180
blocks, 149
failing to block set of statements, 171
GUI applications, 175
nested, 151, 172
if statements, **141,** 180, 288, 497, 591, 593, 619
assignment operator (=), 147–148
Boolean expressions, 141–142
combining AND operator (&&) and OR operator (||), 158–162
comparing variable to valid values, 237
compound expressions, 154–160
conditional AND operator (&&), **154**–156
conditional OR operator (||), **156**–157
as control statement, **144**
equal to operator (==), 147–148
evaluated expression, 141
failing to block set of statements, 171
GUI applications, 175
indenting, 143–144
methods returning value, 142
nested, **145**–146, 155, 163
range checks, 171–172
semicolon (;) incorrectly placed in, 143, 171
IL. *See* intermediate language (IL)
Illustrator class, 461
Image property, 569–570, 592
images, 104
immutable, **79,** 92
implementation hiding, **271,** 282, 302
compromising, 340–341
implicit cast, **71,** 92, 329
implicit conversions, **71**–73, 92, 329, **448**–449, 477
implicitly, **71,** 92
implicit parameter, **366,** 413
implicit reference conversions, **448**–449, 477
INCHES_IN_A_FOOT constant, 81
#include files, 10
incrementing, **194,** 220
indefinite loops, **195,** 199, 220
indenting programs, 12
index, **228**
indexes, 261
IndexOutOfRangeException class, 233, 491, 507, 509–510, 518–519, 528
inFile file, 677
infinite loops, **191,** 194, 220
Infinity value, 492
information hiding, **357,** 412, 434, 438
inheritance, **8,** 39, **424**–426, 461, 477
ancestors, **428**
base class, **427**
child class, **427**
derived class, **427,** 435
extended class, **427**
extending classes, 429–431, 433–434
fields, 428
GUI applications, 473–475
implicit reference conversions, **448**–449
methods, 428
multiple inheritance, 10, **465**
parent-child relationship, 427
parent class, **427**
reusable code, 426
saving time, 426
subclasses, **427,** 465
superclass, **427**
terminology, 427–428
transitive, **428**
initialization, **50,** 90
InitializeComponent() method, 409, 547–550, 573
initializer list, **231,** 261
inner loop, **204,** 220
InputMethodDemo program, 316
InputMethod() method, 316
input streams, 665
instance methods, 78, **355,** 400, 412
delegate encapsulating, 628–629
objects, 361
public access specifier, 357
this reference, 378
instances, **7,** 39, **354,** 412
destroying, 405–407
instance variables, **355**–358, 377, 412
serializable, 684
instantiate, **354,** 412
instantiation, **354,** 412
Int32 class, 354, 470–472, 474
int data type, 48, **56**–57, 59, 71–72, 90, 328–329, 333–334, 552, 668, 672
alias, 50
automatic conversions, 283
integers, **57,** 90
dividing by zero, 492–493, 508
division, 63–64
remainder operation, 63
return type, 298
sorting, 398
subscripts, 230
Integer.TryParse() method, 513
integer variables, 67
integral data types, **56**–57, 90
Integrated Development Environment. *See* IDE (Integrated Development Environment)
IntelliSense, **717,** 720
completion mode, **717**–718
information about methods, 330
parameter information, 719
Quick Info ToolTip, 718

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

IntelliSense (*continued*)
  suggestion mode, **718**
  using statements, 719
InteractiveAddition file, 87–88
interactive programs, **83**–84, 92
InteractiveSalesTax program, 86
interestRate variable, 65
interface keyword, 466
interfaces, **8,** 39, **102,** 130, **398,** 413, 465, **466,** 478
  abstract methods, 466–469, 468
  classes implementing, 468
  creation, 466
  deciding which to use, 122–123
  methods, 399
  names, 399, 466
  polymorphism, 399, 469
intermediate language (IL), **22,** 40
internal access modifier, 356
internal class, 412
internal field modifier, 357
internal keyword, 356
interpreters, 3
int.Parse() method, 86
intrinsic types, **48,** 90, 471
int variables, 68
InvalidCastException class, 491
InvalidDataException class, 491
InvalidOperationException class, 491
InvalidPrinterException class, 490–491
invoking, **5,** 12, 38, **270,** 302
invoking events, **623**
invoking object, **361,** 413
IOException class, 490–491, 515–516, 668
IPlayable interface, 468
isDiscountProvided variable, 69
IsDisposed property, 545
isGoodItem variable, 519
isLocalCustomer() method, 289
isPreferredCustomer() method, 288–289
isProjectUnderBudget variable, 149
isSixMore variable, 70
IssueInvitations() method, 355
isValidIDNumber variable, 141
isValidItem variable, 238, 242
Italic property, 553
itemOrdered variable, 237–238, 240–241
itemPriceAsString variable, 84–85
itemPrice variable, 240–241
Items.Count property, 575
Items property, 572–573
iterations, **190,** 220
iteration variable, **234,** 261
IWorkable interface, 466–469

J

jagged arrays, **256**–257, 261, 296
Java, 5, 10
  ban on passing functions, 613
  declaring array variable, 228
just in time (JIT) compiler, **22,** 40

K

Kaufman, George, 542
Kernighan, Brian, 145
keyboard
  event handling, 639–641
  making selection from, 566
KeyChar property, 640
KeyCode property, 640
KeyData property, 640
KeyDemo program, 641
KeyDown event, 632, 639
key event, 648
KeyEventArgs class, 640
KeyEventArgs parameter, 639
KeyEventHandler delegate, 639
key events, **639**
key field, **663,** 698
KeyPress event, 632, 639
KeyPressEventArgs class, 640
KeyPressEventArgs parameter, 639
KeyPressEventHandler delegate, 639
KeyUp event, 633, 639–641
KeyUp() method, 640
KeyValue property, 640
keywords, **2,** 13, 38
  contextual, **365**
kilobytes, **657,** 697, 715–716

L

label1_Click() method, 121, 555
Label() constructor, 549
Label control, 107, **112**–115, 547, 551
label1 label, 112, 121, 547, 549, 555, 582
label2 label, 582
Label objects, 474, 549, 560, 562–567, 570, 575, 582, 589–590, 594, 608, 633, 635, 640, 642–643, 688, 691
labels, **112,** 122, 124–130, 589, 640
  adding to form, 560–565
  anchoring, 582
  empty, 113–114
  fonts, 117, 590
  multiple lines of text, 112
  names, 112
  positioning, 562
  properties, 560–565
  resizing, 112, 562
  responses to clicks, 638
  text, 112–113
lastName field, 371
LastName property, 372–373
Leave event, 633
LeavesTurnColor() method, 428
left-associative, **705**–706
leftOver variable, 326
Left property, 545
length field, 375
Length property, 79, 233–234, 241, 324, 376, 665, 686
Less than operator (<), 69
Less than or equal to operator (<=), 69
letterButton_Click() method, 643
lexically, **78,** 92
LIMIT constant, 191–192, 194, 200
lincolnCheckBox_CheckChanged() method, 593
lincolnCheckBox object, 587, 588
LincolnForm form, 592
Lincoln Room check box, 593, 597
line comments (//), **18,** 40
LinkClicked() method, 555–577, 633
LinkColor property, 555
link label, **554**–557, 599
LinkLabel class, 554–557
linklabel1_LinkClicked() method, 557
linklabel2_LinkClicked() method, 557
linkLabel1 object, 556
linkLabel2 object, 556
LinkLabel objects, 543, 555–556, 633
LinkVisited property, 555, 557
list boxes, 572, **573,** 599
  automatically selecting item, 576
  displaying items in columns, 574
  items selected in, 576
  number of items in, 575
  objects added to, 573
  position of item in list, 575
  resizing, 574
  selections, 574
  user-selected items, 575
ListBox objects, 543, 572–576, 632
ListControl class, 572
literal characters, 74
literal constants, **48**
literal strings, **11,** 30, 39
Load() method, 555, 574
loanAmount field, 446–447
LoanAmount property, 446–447
Loan class, 432–434, 445, 447, 460
Loan() constructor, 460
Loan objects, 432–434
local, **278**
local variable, 303

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

LocalVariableDemo program, 278
LocationChanged event, 633
Location property, 126, 409, 545, 549, 561–563, 572, 581, 611
logic, **3,** 38
logical errors, 3
  GUI applications, 121
logical operators, 392
logical OR operator (|), 553
logic-planning tools, 138–141
Logo, 4
long data type, **56,** 59, 71–72, 90
  automatic conversions, 283
loop body, **190,** 220
  do loops, 202–203
  executing at least one time, 202
  for loops, 199–200
loop control variables, **191,** 196, 220
  altering by user input, 195
  decrementing, **194**
  incrementing, **194,** 214
  initializing, 199
  partially altering, 196
  starting value, 199
  step value, **199**
  testing, 199
  updating, 199
  while loops, 199
loop fusion, **214,** 220
LoopingBankBalGUI program, 218
LoopingBankBal program, 195–196
LoopingBankBal2 program, 205
loops, **190,** 220
  accumulated totals, **210**–211
  arithmetic performed only once, 213
  avoiding unnecessary operations, 212–213
  combining, 214
  counted loops, **195**
  declaring variable outside, 213
  definite loops, **195**
  do loops, **202**–204
  empty body, **194**
  event-driven programs, 217
  expressions, 213–214
  GUI programs, 216–219
  improving efficiency, 240–241
  improving performance, 212–216
  indefinite loops, **195**
  infinite, **191,** 194
  iterations, **190**
  loop body, **190**
  loop fusion, **214**
  for loops, **199**–201
  nested, 204–209, 292
  not executing, 202
  performing arithmetic on array elements, 233
  placing common items first in, 241
  posttest, **204**
  prefix and postfix incrementing, 214–216
  pretest, **204**
  repeating instructions, 190
  searching arrays, 237–245
  sentinal value, **196**
  short-circuit operators evaluation order, 213–214
  while loops, **190**–198
LostFocus event, 633
lower camel casing, **3,** 38
LOWEST_INVALID_NUM constant, 446
LOW_QUANTITY constant, 161
LOWRATE constant, 208

M

machine language, **2**–3, 38
magic number, **671,** 698
Main() method, 12–13, 21, 26, 28–29, 33, 58, 68, 70, 87, 88, 110–111, 122, 270, 272, 277–280, 291–292, 294, 297–298, 315–316, 323, 341–342, 354, 361, 372, 375, 383, 388, 403, 406, 430, 432–433, 507–508, 524, 529, 613, 617–619, 630, 674–676
majorListBox object, 573
majorListBox_SelectedIndexChanged() method, 575
MakeAppointment() method, 11
MakeDeposit() method, 631
Make property, 447
MakeWithdrawal() method, 631
Mammal class, 448
mandatory parameters, **312**–313, 344
ManyButtons program, 645–647
ManyHellosGUI program, 217
Maple class, 428
Margin property, 545
MarsAlien class, 469
MarshalByRefObject class, 474, 542
Math class, 289
Math.PI constant, 375
Math.Pow() method, 63
matrix, 253
MaxDate property, 577
MAXDINNER constant, 208
MaximumSize property, 583
MAXRATE constant, 208
MaxSelectionCount property, 577
mealButton button, 594, 596
measuring storage, 714–716
megabytes, **657,** 697, 715
MemberAccessException class, 491
memory
  aliases, 314
  limited, 57
  measuring storage, 714–716
  random access memory (RAM), **656**
  volatile, **656**
menu bar, **25,** 40
MenuStrip objects, 585–586
menu strips, 585–586, 599
messageLabel label, 578
Message property, 503, 505, 509, 521, 526
MethodA() method, 521, 529
MethodB() method, 521, 529
method body, **12**–13, 39, 276, 302
MethodC() method, 521, 529
method declaration, **272,** 302
method definition, **272,** 302
MethodGetsArray() method, 294–296
MethodGetsOneInt() method, 294
method header, **12**–13, 39, **272,** 276, 297–298, 302
Method1() method, 337–338
Method2() method, 338–339
MethodN() method, 274
methods, **5,** 38, **270,** 302, 357–358
  abstract classes, 461
  abstract methods, **398**
  accessibility, **273**–274
  ambiguous, **333**–334
  applicable, **329**
  arguments, **11,** 39, 246–247, 270, 276, 281–284
  arrays, 293–296
  automatically generated, 299–300
  betterness rules, **329**
  built-in, 289
  called method, **270**
  calling, **5,** 12, **270,** 279–280
  calling method, **270**
  clients, **271**
  constants, 278, 300
  constructors, **380**–386
  curly braces ({}), 12
  default arguments, 285
  delegates, 612–613
  derived class accessing, 443–444
  encapsulating, 612
  event handlers, **609**–612
  event-handling, 121, 300
  event receiver, **610**
  exception handling thrown from outside methods, 517–520

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

methods (*continued*)
- extension methods, **470**–472
- fully qualified name, **274**
- GUI applications, 299–300
- identifiers, 275–276
- implementation hiding, **271**
- inheritance, 428
- instance methods, 78, **355**
- instructions executed by, 12
- interfaces, **8,** 399
- invoking, **5,** 12, **270**
- jagged arrays, 296
- listing in call stack, 521–525
- local variables, **278**
- mandatory parameters, **312**–313
- manipulating arrays, 245–249
- method body, **12**–13, 39
- method declaration, **272**
- method definition, **272**
- method header, **12**–13, 39, **272**
- multidimensional arrays, 296
- multiple-argument, 284–285
- multiple events, 646–647
- named arguments, **338**–342
- names, 12, 275–276
- nested method calls, **288**
- nonstatic, **274**–275, 300
- no parameters, 272–278
- no return value, 272–278
- not returning value, 13
- objects, 8, 355, 362–363, 612–622
- optional parameters, **312,** 314–321, 335–342
- orphaned, **121,** 131
- output parameters, **314,** 316–320
- overloading, **323**–332, 399
- overload resolution, **329**
- overriding, **399,** 441
- parameter arrays, **314,** 320–321
- parameter list, **276**
- parameters, **276,** 281, 290–292, 303, 312–313
- Pascal casing, 5
- passing by reference, 317
- passing single element to, 293–294
- placing in classes, 276–277
- polymorphism, **9**
- positional arguments, **338**
- private access, 111, **273**–274
- private access modifiers, 374–376
- public access, **273**
- public access modifiers, 373–374
- recursive, **448**
- reference parameters, **314,** 315–316, 322–323
- returning Boolean value, 288–289
- returning value, 142, 247, 275, 286–292
- return type, **275,** 286, 317
- reusable, 272
- signature, **324**
- single-argument, 281–284
- static, 78, 111, **274**–275, 282, 358
- storing, 271, 378
- this reference, **378**–380
- throwing exception and not catching it, 521
- unnamed arguments, 337–338
- valid and invalid declarations, 337
- value parameters, **312**–313
- variables, 278, 300
- virtual methods, **441**
- visible, **442**
- void, 294
- which outside classes can use, 111

MethodS() method, 274
method’s type, **275,** 303
MethodThatAcceptsArray() method, 296
MethodWithItsOwnA() method, 278
Microsoft Sans Serif, 117, 552, 563
Microsoft Visual Studio. *See* Visual Studio
Microsoft Web site, 9
Microsoft.WindowsCE.Forms. DocumentList class, 543
MilesPerGallon program, 492–493
MilesPerGallon2 program, 500–502
MilesPerGallon3 program, 502–503
MilesPerGallon4 program, 504
MinDate property, 577, 579
MINIMUM_LOAN constant, 445
MinimumSize property, 583
mission critical, **494,** 532
M1() method, 612, 615
M2() method, 612, 615
modal window, **591,** 599
ModifierKeys property, 546
modulus operator, 63
monetary values, 60
money variable, 72, 234
monospaced fonts, **117,** 131
MonthCalendar objects, 577–580, 658
MonthlyBoldedDates property, 577
monthly calendar, 577–580
MOTTO constant, 375
MouseButtons property, 546
MouseClick delegate, 639
MouseClick event, 637–639, 638
mouse click triggering default event, 554
MouseDoubleClick event, 637
MouseDown event, 633, 637–638
MouseEnter delegate, 636
MouseEnter event, 633, 635, 637
MouseEventArgs class, 637–638
MouseEventArgs object, 611
MouseEventArgs parameter, 637–638
MouseEventHandler delegate, 637
mouse events, 637–639
MouseHover event, 633, 637
MouseLeave event, 633, 637
MouseMove event, 633, 637
MousePosition property, 546
MouseUp event, 633, 637
MouseWheel event, 633
Move event, 633
Move() method, 658, 660
Movie Discount form, 175
mpg variable, 499–500
MSDN library, 122
MultiColumn property, 572, 574
multidimensional arrays, **252**–256, 261, 296
MultiExtended member, 574
multifile assembly, **271,** 302
multiple-argument methods, 284–285
multiple inheritance, 10, **465,** 478
multiply and assign shortcut operator (*=), 65
MultiSimple member, 574
myAge variable, 50–51
myAssistant object, 359–361, 364
myBook.AdvertisingMessage() method, 378
myCar class, 448
myChef object, 365–366
myDel delegate, 613
myDog class, 448
myGraduationParty identifier, 355
myInteger class, 359
MyMethod() method, 329, 612
myMoney variable, 116
myPetDogFifi object, 428
myScores array, 231, 233–234
myString variable, 84

N

named arguments, **338**–343, 345
named constants, **80**–81, 92, 375, 671
named instances, 552
named variables, 59
name field, 439–441
Name property, 106, 108–109, 113, 118–119, 125, 409, 440–441, 452, 462, 464, 546–547, 549, 556, 561, 573, 592, 594, 608, 611, 665, 688

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

namespace-dot-class-dot-method format, 12
namespaces, **12,** 39
  built-in, 28
  curly braces ({ }), 28
  declaration, 28
  repeatedly using class, 19–20
  user-created, 28
  using clause, **19**
  using directive, **19**
Names.txt file, 674–675, 680
NegativeBalanceException class, 526
nested
  if-else statements, 151, 172
  if statements, **145**–146, 155, 163, 180
nested loops, 204–209, 292
  inner loop, **204**
  outer loop, **204,** 206
nested method calls, **288,** 303
.NET framework culture settings, 62
new field modifier, 357
New Item window, 588
new keyword, 229, 441, 444
newline escape sequence (\n), 74
"Newline in constant" error message, 56
new line (\n), 662
new operator, **229,** 360
New Project command, 32
New Project window, 32, 103, 124, 561
nibbles, **709,** 716
nodes, **104,** 130
no fall-through rule, 165
nonabstract methods, 461, 468
non-application classes, **13**
None member, 574
nonprinting characters escape sequence, **74**
nonstatic methods, **274**–275, 303
nonstatic variables, 358
nonvolatile, 697
Notepad, 21, 557
Not equal to operator (!=), 69, 77
NotImplementedException class, 491
NOT operator (!), **169**–170, 173–174, 181
null escape sequence (\0), 74
NullReferenceException class, 491, 694–695
Number format specifier, 61
numbering systems
  binary numbering system, **708**–709
  decimal numbering system, **707**
  hexadecimal numbering system, **713**–714
numberOfDependents variable, 57
number parameter, 315
numbers
  aligning, 117
  as characters, 73
  converting strings to, 86–87, 317–320
  converting to string, 61–62
  division, 63
  summing, 114
number variable, 147, 191–194, 316, 512–513
numDependents variable, 148
numeric fields, 381
numeric type conversion, 71–73
numeric variables, 57
numOfItems variable, 243–245
numPages field, 377
num1 variable, 153
num2 variable, 153
num3 variable, 153

O

Oak class, 428
Object class, 399, 450–456, 474, 542
Object constructor, 456
object destructors, 10
object initializers, **388**–391
object keyword, 450
object-oriented approach, **6,** 39
object-oriented exception-handling methods, 498–501
  versus traditional exception-handling methods, 497–498
object-oriented programming languages, 7–9
object-oriented programming (OOP), 4, **5**–6, 38
Object parameter, 399, 454
objects, **5**–6, 10, 39, 354, 377, **450**–455, 478
  allocating memory, 360
  arithmetic symbols, 392
  arrays, 230
  attributes, **5,** 7, 39, 355
  behaviors, **5,** 7, 39
  comparing, 453–454
  computer simulations, 6
  constructors, 456
  creation, 359–363, 370–373
  data types, 359–360
  default value, **381**
  deserialization, **684**–687
  encapsulation, **8**
  events, 555
  fingerprints, 455
  GUI applications, 407–410
  GUIs (graphical user interfaces), 6, 102
  identifiers, 359–360
  instance methods, 361
  instantiate, **354,** 424
  interacting with, 608
  interfaces, **8**
  methods, 8, 355
  names, 359–360, 566
  passing to methods, 362–363
  properties, **7**
  reading from text files, 684
  reference equality, **453**–454
  referencing methods, 612–622
  serializable, 684–685
  serialization, **684**–687
  state, **5,** 39, 355
  type or class, 452
  uniquely identifying, 455
  using within another object, 359
  writing to text files, 684
obj folder, 35
Obstetrician class, 431
okButton button, 299, 547
okButton_Click() method, 111, 258, 299
oldEnough variable, 170
OnBalanceAdjusted event handler, 629
OnBalanceAdjusted() method, 629
OnChanged() method, 623
OneButtonTwoEvents program, 635
one-dimensional arrays, **252**–253, 261
  jagged arrays, **256**–257
One member, 574
oneVal variable, 294
Open file mode, 666, 676
opening files, **663,** 698
OpenOrCreate file mode, 666
operands, **63,** 91
  implicitly converting, **71**
Operate() method, 439
operating system error message, 23–24
OperationAbortedException class, 491
OperationCanceledException class, 491
operator*() method, 396
operator+() method, 393–394
operator-() method, 396
operator precedence, **64,** 91, **705**–706
operators
  associativity, **705**
  overloading, 392–396
  precedence, **64,** 91, **705**–706
OptionalParameterDemo program, 336
optional parameters, **312,** 335–342, 344
  output parameters, **314,** 316–320
  parameter arrays, **314,** 320–321
  reference parameters, **314**–315
  unnamed arguments, 337–338

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

order of operation, **64,** 91
OR operator (||), **156**–157, 171, 173, 180, 237
  combining with AND operator (&&), 158–162
orphaned methods, **121,** 131
outer loop, **204,** 206, 220
outFile file, 685, 690
outFile object, 666
out keyword, 314, 317, 319
OutOfMemoryException class, 491
out of scope, **200,** 220
out parameters, 404, 512–513
output
  displaying onscreen, 11–12
  ending with carriage return, 669
  programs producing, 11–13
outputLabel label, 567
outputLabel variable, 216, 219
output parameters, **314,** 316–320, 344
output streams, 665
overloaded binary operators, 392
overloaded constructors, 387–388, 553, 665–666
overloaded operators, 393
OverloadedTriples program, 331–332
overloaded unary operators, 392
overloading, **323,** 345
  constructors, 382–384
  operators, 392–396
overloading methods, 323–332, 399
  ambiguous methods, **333**–334
  betterness rules, **329**
  overload resolution, **329**
  parameter lists, 328
overload resolution, **329,** 345
  named arguments, 342–343
  unnamed arguments, 342–343
override keyword, **461,** 463, 478
override modifier, 441
overrides, **399,** 413
overriding
  abstract methods, 461
  base class members, 439–444
  methods, 441

P

Painter class, 461
Panel class, 584
panels, 583–**584,** 599
parallel arrays, **238,** 243–244, 261
parameter arrays, **314,** 320–321, 344
ParameterDemo1 program, 313
parameterless constructor, **382**–283, 413
parameter lists, **276,** 303
  Main() method, 297–298
  optional parameters, 335–342
  overloading methods, 328
parameters, **276,** 290–292
  actual, **282**
  data types, 281
  formal, **282**
  identifiers, 281
  implicit parameter, **366**
  mandatory, **312**–313
  method header, 276
  methods, 303
  optional, **312,** 314–321, 335–342
  passing to constructors, 381–382
  value, 282
ParamsDemo program, 321
params keyword, 314, 320
Parent class, 468
parent class, **427,** 477
  child class, 430–431, 435
  hiding properties in, **441**
Parent property, 546
parse, 92
Parse() methods, 84, 86–87, 317
partial keyword, 15
PartTimeEmployee class, 426
Party objects, 355
Pascal casing, **5,** 38
PassArrayElement program, 293–294
passed by reference, **294,** 303, 312, 317
passed by value, 312
PassEntireArray program, 295–296
Patent class, 465
path, **657,** 688, 698
PathTooLongException class, 490
payment variable, 65
payRate array, 234
PayRate property, 381
payRate variable, 65, 173
PayrollApplication class, 275
payrollDay variable, 82
pay variable, 4
pepperoniCheckBox object, 566
Percent format specifier, 61
PerformLayout() method, 549
permanent storage, 657
permanent storage devices, **656,** 697
persistent storage, **657,** 698
petabytes, 715
PhoneCall program, 290–292
picture boxes, **569**–571, 599
pictureBox1 object, 572
PictureBox objects, 569–571, 592
Play() method, 439
pointers, 10
Pointer tool, 590
polymorphism, **9,** 39, 439–444
  interfaces, 399, 469
Poodle class, 427
Poodle object, 428
positional arguments, **338,** 345
Position property, 665, 686
postfix decrement operator (--), 66
postfix increment operator (++), **65**–66, 91, 193
  loops, 214–216
posttest loops, **204,** 220
Pow() method, 289
precedence, **705**–706
precision specifier, **61,** 91
PreferredDiscount() method, 617, 621
prefix decrement operator (--), 66
prefix increment operator (++), **65**–66, 91, 193
  loops, 214–216
PrefixPostfixComparison program, 215
pretest loops, **204,** 220
Preview Changes dialog box, 119
price field, 377
priceLabel label, 594–596
PriceListApplication1 program, 518–519
PriceListApplication2 program, 519–520
PriceListApplication3 program, 523
PriceList class, 518–519, 522
PriceList.DisplayPrice() method, 523–524
priceOfCall variable, 291
price parameter, 379
prices array, 238, 242
price variable, 288
priming read, **670,** 698
private access, **273**–274, 276, 303
private access modifiers, 356, 373–376
private class, 412
private field, 436
private keyword, 111
private methods, 111
private modifier, 273–274, 357
procedural programming, 4–6
procedural programs, **4**–5, 38
procedures, 5, 8, 498
Product class, 465
Program class, 28, 33
program comments, **17**–18, 40
  See also comments
Program.cs file, 104, 110

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

programming
  high-level programming languages, **2**–3
  logic, **3**
  object-oriented programming (OOP), 4–6
  procedural, 4–6
programming languages
  assemblers, 3
  camel casing, **3**
  command prompt, 24
  high-level programming languages, **2**–3
  IDE (Integrated Development Environment), 27
  interpreters, 3
  keywords, **2**
  lower camel casing, **3**
  semi-compiled, 22
  syntax, **3**
  syntax errors, **3**
  vocabulary, 3
programs, **2,** 38
  See also GUI applications
  alternative courses, 612
  blocks, **144**–145
  bugs, **3**
  code bloat, **272**
  code refactoring, **118**
  command line, 25
  command-line arguments, 297
  comments, 36
  compilers, **3**
  compiling, 22–30
  debugging, **3**
  deciding environment to use, 29–30
  decision making, 138–141
  design time, **115**
  documenting, 17–18
  events, **608**
  exceptions, **490**–494
  executing, 22–30
  failing to close form before reexecuting, 121
  fault tolerant, **494**
  flowcharts, **138**
  forcing to contain error, 492–494
  IDE (Integrated Development Environment), 22
  indenting, 12
  interactive, **83**–84
  interface, **102**
  intermediate language (IL), **22**
  keywords, 13
  logic, **3**
  logical errors, 3
  logic-planning tools, 138–141
  machine language, 3
  magic number, **671**
  Main() method, 12, 13
  multifile assembly, **271**
  namespace-dot-class-dot-method format, 12
  output, 11–13
  pseudocode, **138**
  reusable modules, 6
  robustness, **494**
  running other programs in, 557
  runtime, **115, 608**
  saving, 22
  self-documenting, **340**
  source code, **22**
  starting point, 12
  statements, 18
  terminating, 515
  text editors, 22
  text editors *versus* IDE (Integrated Development Environment), 28–29
  user input, 87–88
  whitespace, **12**–13
projects
  closing, 127
  forms, 105
  listing files, 104–105
  naming, 103
  saving, 103
prompt, **84,** 92
propagating execeptions, **521,** 533
properties, **7,** 39, **364,** 413, 547, 611
  access modifiers, 365
  accessors, **364**
  auto-implemented property, **368**–369
  automatic properties, **368**–369
  backing field, **365**
  buttons, 109, 126
  creation, 363–369
  curly braces ({ }), 365
  data types, 365
  declaring, 365
  derived classes, 443–444
  description, 105
  expanding or condensing, 106
  forms, 125
  get accessors, **364**–367
  identifiers, 365
  listing of, 105
  naming, 365
  not passed as ref or out parameter, 367
  parent class, 441
  protecting private data, 364
  read from, 365
  read-only, 546
  readonly property, **365**
  read/write, 567
  retrieving field values, 364
  set accessors, **364**–367
  setting fields, 364, 549
  sorting, 105–106
  value acting like keyword, 365
  as variable, 366
  visible, **442**
  written to, 365
Properties folder, 35
Properties window, 103, 109, 121, 125, 546–547, 551, 561
  form name, 105
  listing events, 634–635
  properties, 105–106, 635
proportional fonts, **117,** 131
protected access, **436**–438, 477
protected access modifier, 356, 434–438
protected field modifier, 357
pseudocode, **138,** 179
public access, **273,** 298, 303
public access modifiers, 356, 373–376
public class, 412
public keyword, 289
public modifier, 273–274, 357
public set accessor, 436
purchase variable, 212
PushButtonControl class, 15

Q

Quick Info ToolTip, 718
Quicksort algorithm, 248

R

RadioButton objects, 543, 565–568, 584, 594–596
radio buttons, **565**–568, 594–596, 599
  group boxes, 584
  selected by default, 568
raising events, **608,** 648
RAM. *See* random access memory (RAM)
random access files, **663**
random access memory (RAM), **656,** 697
range checks, **171**–172, 181
range match, **243**–245, 247, 261
RankException class, 491
RATE constant, 440
rateLabel label, 563
rate parameter, 341
rate variable, 65, 292

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

rating variable, 158–160
Read access, 676
reading from files, **657,** 676–677, 697
  sequential access files, 671–673, 677
ReadLine() method, 83–85, 122, 270, 286, 288–289, 317, 665, 667, 672
Read member, 666
Read() method, 84, 665
ReadNameFile program, 676–677
readonly field, 386
readonly modifier, 357, 386
readonly property, **365**
read-only property, 369, 413
read-only variables, **235,** 261
ReadSequentialFile program, 672–673, 677
ReadWrite member, 666
read/write properties, 567
record delimiter, 669, 672
recordIn string, 672
records, **663,** 698
rectangular arrays, **253,** 261
recursive, 477
recursive methods, **448**
redButton button, 645
redButton_Click() method, 646
reference equality, **453**–454, 478
reference parameters, **314**–317, 322–323, 344
reference types, **360,** 412
ref keyword, 314–315, 317
Regular property, 553
Release folder, 35
remainder (modulus) operator (%), 63
remove keyword, 15
Rename command, 118–119
RentFinder program, 255
rents array, 254, 256
RepairValve() method, 431
reportButton button, 112
reserved keywords, 14–15, 30
Resize event, 633
response variable, 196, 202
result variable, 289
ResumeLayout() method, 549
ReThrowDemo program, 530–531
rethrowing exceptions, **529**–531, 533
return statement, **287,** 303
  catch block, 515
  switch structure, 164
return type, **275,** 286, 303, 317
reusable code, 426
ReverseArray program, 249
Reverse() method, 249–252, 270, 294
right-associative, **705**–706
Right property, 546
Ritchie, Dennis, 145
robustness, **494,** 533
root class, **450,** 478
root directory, **657,** 698
Round trip format specifier, 61
rules for betterness, 342
runnable, **13,** 40
runtime, **115,** 130, **176,** 181, **608,** 612, 648
runtime errors, 490

S

salary field, 381, 424–425, 429, 434–436
Salary property, 381, 385, 389, 435
saleAmountForYear variable, 157–158
saleAmount variable, **145**–146, 155, 171–172, 281–282, 285, 617–618
sales array, 229–230, 253
salesArray array, 229
salesCode variable, 173–174
salesList array, 229
salesperson object, 429
salesTable array, 229
sal parameter, 385
SatisfyGraduationRequirements() method, 439
sausageCheckBox_CheckedChanged() method, 567
sausageCheckBox object, 566–567
Save All (Ctrl + Shift + S) keyboard shortcut, 34
SaveMoney frame, 572
saving programs, 22
sbyte data type, **56**–57, 59, 71, 90
  automatic conversions, 283
ScholarshipStudent class, 441, 450–451
Scientific format specifier, 61
scientific notation, **60,** 91
scope, **278,** 303
  GUI applications, 300
scores array, 250–252
score variable, 86, 318–319
screen, displaying output, 11–12
ScrollableControl class, 474
Sculptor class, 461
sealed classes, **470,** 478
Search command, 23
searching arrays
  arrays of objects, 398–402
  binary search, **246**–247
  for loops, 238–240
  with loops, 237–245
  parallel arrays, **238**
  range match, **243**–245
  sequential search, **238**
  while loops, 241–242
secondDel delegate, 613, 615, 617, 619
seekEmp object, 401–402
Seek() method, 675, 677–683
SeekOrigin enumeration, 677–678
SelectedIndexChanged() method, 572
SelectedIndex property, 572
SelectedIndices property, 572
SelectedItem property, 572–573, 575
SelectedItems property, 572, 576
SelectionEnd property, 577–578
SelectionMode enumeration, 574
SelectionMode property, 572, 574
SelectionRange property, 577
SelectionStart property, 577–578
Select Resource window, 570, 592
self-documenting, **340,** 345
self-documenting statements, **81,** 92
semantic errors, 38
semicolon (;) expected error message, 120
sentinal value, **196,** 220, 671
sequence structure, **138,** 180
sequential access files, **663,** 698
  reading from, 671–673
  rereading, 677
  searching, 677–680
  writing data to, 668–671
sequential search, **238,** 261
[Serializable] attribute, 684
SerializableDemonstration program, 685–687
serialization, **684**–687, 698
Serialize() method, 685
set accessors, **364**–369, 371, 373, 413, 429, 435, 440, 446–447, 527, 623
set_AccountNum set assessor, 528
set_Balance set assessor, 528
SetIDNumber() method, 363–364
set keyword, 15
SetPriceAndTax() method, 379
SetSelected() method, 576
setter, **365,** 413
Shift property, 640
short-circuit evaluation, **156**–157, 180, 289
short-circuit operators, 213–214
shortcut arithmetic operators, 64–66
  add and assign shortcut operator (+=), **65**
  divide and assign shortcut operator (/=), 65

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

  multiply and assign shortcut operator (*=), 65
  postfix arithmetic operator (++), **65**–66
  postfix decrement operator (--), 66
  prefix decrement operator (--), 66
  prefix increment operator (++), **65**–66
  subtract and assign shortcut operator (-=), 65
shortcut operators, 193
short data type, **56**–57, 59, 71, 90, 283
ShowDialog() method, 591
ShowDog class, 8–9
ShowTodayCircle property, 577
ShowToday property, 577
ShowWelcomeMessage() method, 276–277
side effects, **157**–158, 180
signature, **324,** 345
significant digits, **59,** 91
simple data types, **48,** 90, 360
SimpleMethod() method, 333–334
single-argument methods, 281–284
single-dimensional arrays, **252,** 261
single quotation mark escape sequence (\'), 74
SizeMode property, 569
Size property, 105–106, 109, 112, 125, 409, 546, 549, 572, 608, 611, 645
snap lines, **581,** 599
SoapException class, 490
software, **2,** 38
Solution Explorer, 103, 409, 564, 593
  files part of current project, 104–105
  form name, 105
  nodes, **104**
  project name, 103
someMoney variable, 52–54
someNums array, 293
SomeOtherClass class, 277
someValue variable, 65
someWorker object, 452
SortArray program, 248
Sorted property, 572
sorting
  arrays, 246–248
  arrays of objects, 398–402
  characters, 398
  integers, 398
  properties, 105–106
Sort() method, 248, 250–252, 270, 294
source code, **22,** 40
  colors, 720
  syntax errors, 24
Speak() method, 462–464
specialButton button, 177–178
specialButton_Click() method, 178
Split() method, 672
Spruce class, 427–428
Spruce objects, 428
StackOverflowException class, 491
StackTrace property, 521–526, 528
StandardDiscount() method, 617, 621
standard numeric format strings, **60**–61, 91
StartsWith() method, 80
state, **5,** 39, 355
statements, 18
  commenting out, **17**
  ending with semicolon (;), 11
  executing two or more conditionally, 144
  method body, 276
  self-documenting, **81**
static classes, 475
static field modifier, 357
static keyword, **13,** 20, 40, 111, 274–276, 280, 289, 300, 375
static methods, 78, 111, **274**–275, 282, 303, 358, 394, 471, 621
static modifier, 358
STEP keyword, 199
step value, **199,** 220
Stopwatch class, 214–215
StreamReader class, 665
StreamReader objects, 671, 676, 682, 692
streams, **664**–668
  exposing, **665,** 698
  input, 665
  output, 665
StreamWriter class, 665, 689
StreamWriter constructor, 666
StreamWriter object, 666, 669, 674–675
Strikeout property, 553
string argument, 552, 557, 613
String class, 77, 398, 453, 474, 672
String Collection Editor, 573
string data type, 48, **76**–80, 92, 286, 669, 671–672
String.Format() method, 53, 116–117
string interpolation, 53
String objects, 453
string objects, 676, 684, 692
strings, 76–80, 324
  accessing character with subscripts, 232
  arguments, 11
  assigning array character to, 231
  characters, 74
  combining values, 53
  comparing, 77–79, 453
  concatenating, **53,** 90, 392
  console input, 84
  converting, 84–88, 512–513
  converting integer to, 325
  converting numbers to, 61
  converting to number, 86–87, 317–320
  data types, 85–86
  delimiters, **669**
  escape characters, 117
  format string, **53**–54
  formatting, 53–54, 116–117
  forms, 106
  holding name of class, 452
  immutable, **79**
  interpreting characters literally, 557
  largest and smallest values, 48
  length, 79, 234
  lexically, **78**
  placeholders for values, 53
  separating fields into array of, 672
  starting characters, 80
  substrings, 79
  text boxes, 114
  tokens, **669**
  user input, 88
struct, 360
StudentChanged() method, 624
Student class, 370–373, 387, 439–441, 450–451, 622–627, 632, 663
Student constructor, 403
Student objects, 372–373, 403, 624
stu object, 623
subclasses, **427,** 461, 465, 477
subroutines, 5
subscripts, **228,** 230, 261, 675
  accessing character in array, 232
  constants, 233
  integers, 230
  range of, 233–234
  two-dimensional arrays, 253
  variables, 233
Substring() method, 79
substrings, 79
subtract and assign shortcut operator (-=), 65, 615
subtraction operator (-), 63, 615
sub variable, 233
SUDSParserException class, 490
suggestion mode, **718,** 720
sumButton button, 118
summing numbers, 114
sum property, 114
superclass, **427,** 461, 477
SuspendLayout() method, 549
Swap() method, 322–323

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

SwapProgram program, 322–323
switch-case structure. *See* switch structure
switch keyword, 164
switch structure, **163**
- alternative courses of action, 166
- break keyword, 164–165
- case keyword, 164
- case labels, **164**
- default case, 165
- default keyword, 164
- enumerations, 166–167
- governing type, **164**
- GUI applications, 175, 177–178
- return statement, 164
- switch keyword, 164
- throw statement, 164

syntax, **3,** 38
syntax errors, **3,** 24, 38, 120
- IDE (Integrated Development Environment), 27

System.Array class, 230, 234, 245, 398
System.Console class, 279
System.Diagnostics.Process.Start() method, 557
SystemException class, 492–494, 494
SystemException objects, 495–497
System.IO namespace, 658, 692
System namespace, 12, 18–20, 450, 471, 515
- aliases, **48**
- fully qualified data type names, 50

System.Object class, 450
System.Runtime.Serialization.Formatters.Binary namespace, 684
System.Runtime.Serialization namespace, 684
system software, **2,** 38
System.Windows.Forms.AxHost class, 543
System.Windows.Forms.ButtonBase class, 543
System.Windows.Forms.Button class, 610
System.Windows.Forms.Control class, 543–544
System.Windows.Forms.DataGrid class, 543
System.Windows.Forms.DataGridView class, 543
System.Windows.Forms.DateTimePicker class, 543
System.Windows.Forms.GroupBox class, 543
System.Windows.Forms.Integration.ElementHost class, 543
System.Windows.Forms.Integration.Label class, 543
System.Windows.Forms.Integration.ListControl class, 543
System.Windows.Forms.Integration.ListView class, 543
System.Windows.Forms.Integration.MdiClient class, 543
System.Windows.Forms.Integration.MonthCalendar class, 543
System.Windows.Forms.Integration.PictureBox class, 543
System.Windows.Forms.PrintPreviewControl class, 544
System.Windows.Forms.ProgressBar class, 544
System.Windows.Forms.ScrollableControl class, 544
System.Windows.Forms.ScrollBar class, 544
System.Windows.Forms.Splitter class, 544
System.Windows.Forms.StatusBar class, 544
System.Windows.Forms.TabControl class, 544
System.Windows.Forms.TextBoxBase class, 544
System.Windows.Forms.ToolBar class, 544
System.Windows.Forms.TrackBar class, 544
System.Windows.Forms.TreeView class, 544
System.Windows.Forms.WebBrowserBase class, 544

T

TabIndex property, 116, 546, 549, 645–646
TabIndex value, 642, 645–646
tables, 253
tab order, **116,** 131
TabStop property, 116, 642
TakeBloodPressure() method, 431
Tax class, 521–522, 524
Tax.DetermineTaxRate() method, 524
taxRate array, 522
TAX_RATE constant, 81
taxRate variable, 285
temporary variables, 323
tempTotal variable, 213
temp variable, 401
terabytes, 715
terminating programs, 515
ternary operators, 168, 180
TestCarpet class, 375–376
testing for equivalency, 147–148
testScore variable, 169
TextBox class, 113, 216
TextBox control, **112**–117
text boxes, **113**–115, 122, 130, 176, 330
TextBox objects, 544, 688–691
TextChanged event, 633
Text control, 474
Text Editor, 121
text editors, 21
- filename extensions, 22
- find and replace feature, 234
- *versus* IDE (Integrated Development Environment) and, 28–29
- identifiers, 28

textExpression variable, 168
text files, **656,** 697
- GUI application creation, 688–691
- reading data, 683–687
- reading data into form, 691–696
- reading objects from, 684
- writing to, 683–687

text in labels, 112–113
Text property, 105–106, 109, 112–114, 125–127, 216, 219, 546–547, 549, 555, 561–562, 565, 567, 569–570, 572, 575, 587, 589, 592, 594–595, 608, 610, 632–633, 635, 642–643, 645, 688, 691, 694–695
TextReader class, 665
TextWriter class, 665
theTreeInMyBackYard object, 428
thirdDel delegate, 619
this keyword, 471, 548
this reference, **378**–380, 385, 400, 413
three-dimensional arrays, 256
ThreeLinesOutput class, 28
ThreeLinesOutput.cs file, 23–25
ThreeLinesOutput.exe file, 25
ThreeLinesOutput project, 27
throwing exceptions, 498–503, 505–506, 519, 528–529
throw keyword, 529, 531
- switch structure, 164

tickets array, 257
ticketsUnderMinimum variable, 170
time, 577–580, 658
TIMES constant, 214
TippingTable program, 207–209

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

TippingTable2 program, 210
tipRate variable, 208–209
TIPSTEP constant, 208
title field, 377
ToBoolean() method, 86
ToByte() method, 86
ToChar() method, 86
TodayDate property, 579
ToDecimal() method, 86
ToDouble() method, 84, 86
ToInt16() method, 86
ToInt32() method, 86, 513
ToInt64() method, 86
tokens, **669,** 698
ToLongDateString() method, 578
Toolbox, 126, 562
    adding button to form, 107–112
    listing controls, 107
    Menus & Toolbars group, 585
    pinning to screen, 107
TopLevelControl property, 546
TOPPING_PRICE constant, 567
Top property, 546
ToSbyte() method, 86
ToShortDateString() method, 578
ToSingle() method, 86
ToString() method, 86, 299, 372, 451, 502–503, 573, 610
    format specifiers, 61
    overriding, 452
total array, 258
totals
    accumulated, **210**–211
    add and assign shortcut operator (+=), 211
total variable, 211–212
ToUInt16() method, 86
ToUInt32() method, 86
ToUInt64() method, 86
traditional exception-handling methods *versus* object-oriented exception-handling methods, 497–498
TRANSACTIONS constant, 631
transitive, **428,** 477
Tree class, 427–428
triggering events, **608,** 648
Triple() method, 331–332
true-or-false comparisons, 68–69, 140, 142, 180
Truncate file mode, 666
truth tables, **154,** 180
TryBankAccount program, 527–528
try block, **498**–502, 505–511, 514–516, 519, 533, 668
try...catch block, 695
try keyword, 498
TryParse() method, 318–320, 367, 512–513
tuition field, 439–441
Tuition property, 440
two-dimensional arrays, **252**–254
TwoErrors class, 507–508
TwoErrors program, 507–508
TwoErrors2 program, 508–509
TwoErrors3 program, 509–510
.txt file extension, 557
type precedence, **71,** 92
type-safe data types, **83,** 92

U

uint data type, **56,** 71–72, 90
    automatic conversions, 283
ulong data type, **56,** 71, 90, 283
UML. *See* Unified Modeling Language (UML) diagrams
unary form, 392
unary operators, **66,** 91, 168
    overloaded, 392, 396
    right-associative, **705**
Underline property, 553
UnhandledException window, 694
Unicode, **48,** 74–75, 90, 662, **709,** 716
    characters represented as numbers, 56
    foreign alphabets, 75
    letter values, 106
Unified Modeling Language (UML) diagrams, **426,** 477
unifying type, **71,** 92
uninitialized variables, 51
unnamed arguments, 337–338
    overload resolution, 342–343
unnamed constants, 48
unreachable, **510,** 533
unsafe mode, 10
upper camel casing, **5,** 38
user-created namespaces, 28
user-defined application exception classes, 492
UseTaxMethod program, 283
ushort data type, **56**–57, 59, 71, 90
    automatic conversions, 283
using clause, **19,** 40
using directive, **19,** 40
using statements, 21, 28–29, 33, 616, 719
using static System.Console statement, 20, 26, 28

V

ValidID project, 197–198
validValues array, 237–238, 240–241
ValueChanged property, 580
value keyword, 15, 366
value parameters, 282, **312**–313, 335, 344
Value property, 580
value-returning methods, 275, 286–289
values
    comparing to valid values in array, 237
    garbage, **211**
value types, **360,** 412
value variable, 70
Vampire class, 469
variable declaration, **50,** 90
variables, **4,** 8, 38, **48,** 90
    aliases, 314–315
    arrays, 228
    assigning values, 50–51
    camel casing, 3
    checking value, 497
    combining values, 53
    comparing to valid values, 237
    data types, 50, 57
    declaring, 48–51, 57–59
    displaying values, 52–56
    enumerations, 82
    fixed number of integral values for, 82
    identifiers, **4,** 50
    increasing value in, 64–66, 193
    initializing, **50**
    initial value, 50
    instance variables, **355**
    local, **278**
    methods, 278, 300
    naming, 49–50
    nonstatic, 358
    out of scope, **200**
    postfix arithmetic operator (++), 65
    prefix arithmetic operator (++), 65
    read-only, **235**
    reducing value in, 66
    scope, **278**
    subscripts, 233
    uninitialized, 51
    usage, 58–59
variable substitution, 53
verbatim identifiers, **14,** 40
vertical tab escape sequence (\v), 74
viewButton button, 691
viewButton_Click() method, 692, 695
ViewInvoices program, 691–696

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

virtual methods, **441,** 461, 477
visible, **442,** 477
VisibleChanged event, 633
Visible control, 474
Visible property, 106, 109, 128, 546
VisitedLinkColor property, 555
Visual Basic, 5
Visual Studio, 9
- automatic completion, 29
- closing, 35
- compiling code with IDE (Integrated Development Environment), 25–28
- editor, 21
- enumerations, 82
- Help documentation, 122
- namespaces, 12, 28
- program comments, 36
- Properties window, 546
- using statements, 28

Void class, 16
void keyword, **13,** 40, 276
void methods, 275, 294
void return type, 297, 613
volatile, 697
volatile field modifier, 357
volatile memory, **656**

W

wasSuccessful variable, 513
WEDNESDAY constant, 83
weeklyBudget variable, 72
welcomeLabel label, 562
WelcomeMessage() method, 358, 361, 365
where keyword, 15
while keyword, 190
while loops, **190**–198, 216, 220, 519, 670, 672, 682
- comparing value to valid values in array, 237
- curly braces ({ }), 193
- definite loops, 199
- displaying integers 1 through 10, 199
- empty body, **194**
- indefinite loops, 199
- loop body, 190–191
- loop control variable, **191,** 199
- navigating arrays, 234–235
- nested, 204
- performing arithmetic, 233
- searching arrays, 241–242

whiteButton button, 645
whitespace, **12**–13, 40
whole numbers, 56–57
widgets, **542,** 599
width field, 375
Width property, 106, 376, 390, 544, 546
Width value, 583
Window objects, 585
windows, 107
WindowsFormsApplication1 form, 102–103
worker1 object, 401
worker2 object, 401
Work() method, 466–468
WriteLine() method, 11–12, 26, 33, 58, 61–62, 66–68, 76, 84, 122, 149, 208, 270–272, 274, 281, 284, 288–289, 324, 434, 460, 500, 524, 665–666, 669, 671–672
- displaying variable values, 52–56

Write member, 666
Write() method, 12, 84, 665–666
- displaying variable values, 52–56

writer objects, 666, 669, 690
WriteSequentialFile class, 670–671
WriteSomeText program, 667
writing to files, **657,** 698

X

XML-documentation format contents (< >), **18,** 40
XML (eXtensible Markup Language), 18, **668**
XML files, 668
XmlTextReader class, 668
XmlTextWriter class, 668
X property, 638
x variable, 313, 315–316
XxxException placeholder, 499

Y

Year property, 446
yearsOfService variable, 157–158
year variable, 163, 165
yesButton_Click() method, 219
yield keyword, 15
yottabytes, 715
yourAge variable, 51
yourAnniversaryParty identifier, 355
yourBook.AdvertisingMessage() method, 378
Y property, 638

Z

zeroth element, 230
zero variable, 495
zettabytes, 715
ZONE1 constant, 161
ZONE2 constant, 161

Copyright 2016 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.