

# Index

## A

**abstract classes, 196–197, 210, 233–235**

**accessors, 244**

with nonpublic accessibility, 261

**ActiveX Data Objects.NET (ADO.NET), 9, 933–989. See also**

**DataSet object; System.Data namespace**

database updating, 952–965

rows, 956–964

DataReader, 940–948

DataSet, 948–952

access options, 948–952

data fill, 948

multiple table access, 965–973

support, 973

definition, 934

design goals, 935–936

extensibility, 936

multitier application support, 936

relational data access, 936

XML unification, 936

history, 935

LINQ over Dataset, 985–989

applications, 985

nomenclature, 934

objects, 937–939

consumer, 938–939

provider, 937–938

System.Data namespace, 939–940

SQL support, 977–985

data adapter commands, 977–981

stored procedures, 983–985

where clause, 978

XML, 973–977

DataSet support, 973

**ADO.NET. See ActiveX Data Objects.NET**

**Advanced Research Projects Agency Network (ARPANET), 710**

**aggregate operators, 865–868**

numeric, 865–868

properties, 865

**ajax.asp.net, 740**

**alignment control, 1175**

**All(), 877–878**

**anchoring control, 435–437**

properties, 437

**animations, 1191–1192**

timelines, 1192–1194

with keyframes, 1192–1193

without keyframes, 1193–1194

**anonymous methods, 388**

lambda expressions, 418–423

**anonymous types, 408–411**

application, 409–410

execution, 410–411

**anonymous users, 687–688**

events, 687

**Any(), 877–878**

**appearances editor, 693**

**application state, ASP.NET, 650**

**application-specific web services, 713**

**apply button, 578–579**

**ARPANET. See Advanced Research Projects Agency Network**

**arrays, 93, 111–118**

application, 113

collections v., 200, 277–285

application, 277–281

code modification, 279–281

execution, 282–285

declaration of, 111–112

elements, 111

execution, 113–114

foreach loops, 114–116

initialization, 112

jagged, 116–117

multidimensional, 115–116

parameter, 133–135

application, 134

single base types, 111

size matching, 112

**as operators, 327–328**

**ASP.NET, 627–669. See also Asynchronous Javascript and XML;**

**deployment, of web applications; state management,**

**ASP.NET**

AJAX, 739–757

client script extensions, 742

controls, 741, 746–749

overview, 739–741

update panels, 741–746

web services, 749–754

authentication/authorization, 652–660

configuration, 652–657

CreateUserWizard, 659–660

login page creation, 658–659

membership provider, 652

security controls, 657

event handlers, 637–641

PreviousPage creation, 640–641

results page creation, 639–640

HTML code, 628–637

HTTP, 628

input validation, 642–646

controls, 642–645

overview, 628

runtime, 628

IIS, 628

server controls, 636–637

SQL server database, 660–669

## ASP.NET (continued)

- data controls, 662
- database creation, 660–661
- state management, 646–651
  - client-side, 646–648
  - server-side, 648–651
- user input display, 638–639
- web application deployment, 759–772
  - copying websites, 763–765
  - IIS, 628
  - publishing websites, 765–767
  - Windows Installer, 767–772
- web services, 720–722
  - client implementation, 731–732
  - method addition, 721–722
- web site creation, 629–636
  - control properties, 630–631
  - partial keyword, 635
- assemblies**
  - GAC, 5
  - meta information, 5
  - MSIL code, 5
  - resources for, 5
- assembly attributes, 1020–1023**
- AssemblyInfo.cs file, 22**
- assertions, 170–171**
  - Boolean value, 170
- assignment operators, 51–52**
  - Boolean, 66
  - variables types, 40
- assignments, 36, 45–46**
- Asynchronous Javascript and XML (AJAX), 739–757**
  - client script extensions, 742
  - controls, 741, 746–749
    - extender, 755–757
    - timer, 746–747
    - update progress, 747–749
  - overview, 739–741
  - update panels, 741–746
    - properties, 746
    - ScriptManager, 743
    - triggers, 744–746
  - web services, 749–754
  - WCF, 749–754
- asynchronous service calls, 728–731**
- attached event, 1173**
- attached properties, 1168**
- attribute syntax, 1161**
- attributes, 1019–1046**
  - assembly, 1020–1023
  - built-in, 1026–1027
  - custom, 1038–1046
    - AttributeUsage.AllowMultiple, 1043–1045
    - AttributeUsage.Inherited, 1046
    - BugFixAttribute, 1039
    - class usage, 1040–1041
    - constructors, 1040
    - properties, 1040
    - scope, 1042–1043
    - System.AttributeUsageAttribute, 1041–1042
  - definition, 1019–1023
  - reflection, 1023–1026
  - syntax, 1161–1162
  - System.Diagnostics.ConditionalAttribute, 1027–1029
  - System.ObsoleteAttribute, 1029–1031

- System.Reflection.AssemblyDelaySignAttribute, 1034–1038
- delay signing, 1035–1037
- GAC registering, 1037–1038
- strong name completion, 1038
- System.SerializableAttribute, 1031–1034
- AttributeUsage.AllowMultiple, 1043–1045**
- AttributeUsage.Inherited, 1046**
- authentication/authorization, ASP.NET, 652–660**
  - configuration, 652–657
  - CreateUserWizard, 659–660
  - login page creation, 658–659
  - membership provider, 652
  - security controls, 657
- auto-completion, 120–123**
  - IDE, 123
- automatic properties, C# language, 253**

## B

- BackColor, 435**
- behavior editor, 693**
- binary operators, 46–48, 52**
  - overloading, 312
- binary serialization, 686**
- bindings, 1216, 1218–1220**
- bitwise operators, 62–67**
  - application of, 67
  - shift, 65–66
  - Two's complement, 64
- bitwise shift operators, 65–66**
- Boolean logic, 59–69**
  - goto statement, 69–70
  - spaghetti code, 69
  - operators, 46, 61–69
    - application of, 66–67
    - assignment, 66
    - bitwise, 62–67
    - conditional, 61–62
    - precedence updates, 68–69
    - values, 60–61
- Boolean values, 36, 38**
  - assertions, 170
  - LINQ, 877–878
  - All(), 877–878
  - Any(), 877–878
- border control, 1175**
- Bottom, 435**
- bottom property, 435**
- bound data, updating of, 928–929**
- boxing, 304–305**
- branching, 59, 70–78**
  - if statements, 71–75
    - application, 72–74
    - conditions, 74–75
  - switch statements, 75–78
    - application, 77–78
    - break statements, 75–76
    - constant variables, 77
    - default statements, 76
    - execution, 78
    - structure, 75–76
  - ternary operators, 46, 70–71
  - variable scope, 141
- break mode, debugging, 166–176**

- breakpoints, 167–170
    - addition of, 168
  - Command windows, 175–176
  - entering, 166–171
    - assertions, 170–171
  - Immediate windows, 175–176
  - break statements, 75–76**
  - infinite looping, 90
  - breakpoints, 167–170**
    - addition of, 168
    - drop-down list, 169
    - VS 2008, 168
  - broadcasting, for networks, 1075**
  - brush class, in GDI+ class, 1122–1124**
    - texture, 1131–1133
  - BugFixAttribute, 1039**
  - building block web services, 713**
  - built-in attributes, 1026–1027**
  - BulletIndent, 462**
  - button control, 440–443**
    - events, 441–443
    - handlers, 442–443
    - properties, 440
  - byte conversions, 107**
- C**
- C# language, 8. See also ActiveX Data Objects.NET; class members, C# language; functions, for C#; Language Integrated Query; object-oriented programming techniques; System.Object; 3.0 enhancements, C# language; variables; Visual C# 2008 Express Edition; Visual Studio 2008; Windows Forms Designer**
  - ADO.NET, 9, 933–989
    - database updating, 952–965
    - DataReader, 940–948
    - DataSet, 948–952
    - definition, 934
    - design goals, 935–936
    - history, 935
    - LINQ over Dataset, 985–989
    - nomenclature, 934
    - objects, 937–939
    - SQL support, 977–985
    - XML, 973–977
  - applications, 8–9
    - for web and services, 9
    - Windows, 8
  - as block-structured, 32
    - nesting, 32
  - C++ v., 8
  - class definitions, 195–197, 209–238
    - abstract, 196–197, 210, 233–235
    - inheritance, 195–197
    - interfaces, 212–215, 233–235
    - internal, 210
    - modifiers, 212
    - objects, 187
    - parent, 195
    - sealed, 197, 210
    - virtual, 196
  - class members, 241–276
    - application, 247–248
    - class diagram, 249–252
    - definitions, 241–242
    - execution, 248–249
    - fields, 242
    - internal, 241
    - methods, 242–244, 250–251, 254–256
    - nondefault constructors, 249
    - private, 241
    - properties, 244–247
    - protected, 241
    - public, 241
    - refactoring, 252–253
    - this keyword, 256–257
  - debugging, 155–176
    - break mode, 166–176
    - call stack window, 176
    - configurations, 156
    - directories, 156
    - nonbreak mode, 157–166
    - stepping through code, 174–175
    - symbolic information, 156
    - variable content monitoring, 171–174
  - error handling, 155, 176–184
    - configuring exceptions, 182
    - definition, 176
    - fatal, 155
    - listing exceptions, 182
    - logic/semantic, 155
    - namespaces, 177
    - SEH, 177–182
  - flow control, 59–90
    - Boolean logic, 59–69
    - branching, 59, 70–78
    - goto statement, 69–70
    - looping, 59, 78–90
  - functions, 125–154
    - code, 125, 127
    - command-line arguments, 126, 145–147
    - data exchange applications, 130–131
    - definition, 126–138
    - delegates, 126, 150–153
    - execution code, 131–132
    - Main(), 144–147
    - overloading, 126
    - parameters, 130–138
    - PascalCase, 42
    - purpose, 125
    - signatures, 126, 149
    - struct, 147–148
    - use of, 126–130
    - variable scope, 138–144
  - generics, 331
  - interfaces, 212–215, 233–235
    - lollipop syntax, 193
  - LINQ, 9, 417–429, 849–892
    - aggregate operators, 865–868
    - Boolean values, 877–878
    - complex objects, 868–872
    - First(), 886–887
    - first query, 850–854
    - FirstOrDefault(), 886–887
    - group queries, 881–883
    - joins, 891–892
    - lambda expressions, 418–429, 856–858
    - large data sets, 862–864
    - methods, 855–856

## C# language (continued)

- orderby clause, 860
- ordering in, 860–862
- projection, 872–875
- query results, 858–859
- resources, 892
- select distinct query, 875–876
- set operators, 887–891
- Skip(), 883–885
- Take(), 883–885
- 3.0 enhancements, 417
- varieties, 850
- methods
  - addition of, 250–251
  - anonymous, 388, 418–423
  - definition, 242–244
  - extension, 412–417
  - hidden, 254–256
  - ListBox controls, 471
  - ListView control, 477
  - override, 243, 254–256
  - partial, 264–265
  - .NET Framework compatibility, 4
- OOP techniques, 4, 185–206
  - components, 187–191
  - definition, 186–192
  - functional, 186
  - inheritance, 195–197
  - instance class members, 191–192
  - interfaces, 193–194
  - operator overloading, 201–202
  - polymorphism, 197–199
  - reference v. value types, 202
  - static members, 191–192
  - structure, 186
  - UML, 187, 192–193, 195, 200
  - Windows applications, 203–206
- syntax, 32–34
  - case-sensitivity, 34
  - comments, 33
  - indentations, 32
  - statements, 32
  - white space characters, 32
- System.Object, 211, 215–218
  - methods, 216–217
- 3.0 enhancements, 399–429
  - anonymous types, 408–411
  - automatic properties, 253
  - extension methods, 412–417
  - initializers, 400–406
  - lambda expressions, 418–429
  - partial methods, 264–265
  - type inference, 406–407
- as type-safe, 8
- variables, 36–56, 93–123
  - arrays, 93, 111–118
  - assignments, 36, 45–46
  - constant, 77
  - declaration of, 45–46
  - enumerations, 93, 103–118
  - expressions, 46–56
  - literal values, 43–45
  - namespaces, 53–56
  - naming, 41–43
  - operators, 46–53
    - string manipulation, 118–123
    - structs, 93, 108–110
    - types, 36–41
- Windows Forms Designer, 13, 24–28, 433–491
  - code outlining, 27–28
  - controls, 434–491
  - creation, 25–26
  - execution, 26–28
  - Form1.cs file, 27–28
  - Form1.Designer code, 27–28
  - OOP techniques, 203–206
  - VCE, 25–28
  - VS 2008, 13, 25–28
  - WPF, 1148
  - writing programs, 13–29
    - console application, 13, 18–21, 34–35
- C++ language, 32**
  - C# v., 8
  - generics, 331
  - .NET Framework compatibility, 4
- caches, 651**
- call stack window, 176**
- camelCase, 42**
- CanRedo, 462**
- CanUndo, 462**
- canvas control, 1175–1176**
- CardLib, 229–233**
  - application, 230–232, 269–274
  - console, 274
  - collections, 302–303
    - deep copying, 302–303
  - custom exceptions, 374–375
  - events, 389–397
    - application, 389–397
    - expansion, 389
  - execution, 232–233
  - generics, 351
  - operator overloading, 314–320
- catalog zone, 695–697**
  - addition of, 697
- catch, as keyword, 177–178**
- CausesValidation, 445**
- certifications, 586**
- char types, 93–94**
- CharacterCasing, 445**
- CheckBox control, 454, 456–457**
  - events, 456
  - GroupBox control, 457–458
  - properties, 456
- CheckedListBox control, 469–474**
- class definitions, C# language, 195–197, 209–238**
  - abstract, 196–197, 210, 233–235
  - constructors, 218–222
    - execution sequence, 219–222
  - copying, 238
    - deep v. shallow, 238
  - destructors, 218–219
  - diagrams, 227–229
  - inheritance, 195–197
  - interfaces, 193–194, 212–215, 233–235, 257–261
    - accessors, 261
    - application, 213–214
    - execution, 214–215
    - explicit, 260–261
    - implementation, 257–261

- implicit, 260–261
- lollipop syntax, 193
- multiple, 194
- polymorphism, 198–199
- internal, 210
- library projects, 229–233
  - application, 230–232, 265–274
  - execution, 232–233
- modifiers, 212
- objects, 187
- parent, 195
- partial, 262–263
- sealed, 197, 210
- System.Object, 211, 215–218
  - methods, 216–217
- virtual, 196
- class diagrams, 227–229**
  - XML documentation, 1059–1063
- class library projects. See CardLib**
- class members, C# language, 241–276**
  - application, 247–248
  - class diagrams, 249–252
  - definitions, 241–242
    - nesting types, 257
  - dialogs, 537–581
    - ColorDialog, 579–580
      - file, 540–559
      - FolderBrowserDialog, 580–581
      - FontDialog, 577–579
      - general applications, 539–540
      - PageSetupDialog, 568–570
      - PrintDialog, 570–574
      - printing, 559–577
      - PrintPreviewDialog, 575–576
      - public instance methods, 538
  - execution, 248–249
  - fields, 242
    - addition of, 252
  - interfaces, 257–261
    - implementation, 257–261
  - internal, 241
  - methods, 242–244
    - addition of, 250–251
    - hidden, 254–256
    - override, 243, 254–256
  - nondefault constructors, 249
  - private, 241
  - properties, 244–247
    - addition of, 251–252
    - automatic, 253
  - protected, 241
  - public, 241
    - PascalCase, 243
  - refactoring, 252–253
  - this keyword, 256–257
- Class View window, 223–225**
  - symbols, 224
- Click, 438**
- ClickOnce deployment, 584–597**
  - certifications, 586
  - manifest signing, 585–586
  - permission requirement definitions, 587–589
  - publish configuration options, 589–590
  - publish wizard, 590–597
    - application, 594–597
    - updates, 595–597
- client application types, web services, 713**
- client-side state management, 646–648**
  - cookies, 647–648
  - ViewState, 647
- clipping, 1138–1139**
- CLR. See Common Language Runtime**
- COBOL language, .NET Framework compatibility, 4**
- code nesting, 32**
- code outlining, 27–28**
- collection initializers, 402–406**
  - application, 403–404
  - execution, 405–406
- collections, 200–201, 277–303**
  - arrays v., 200, 277–285
    - application, 278–281
    - code modification, 279–281
    - execution, 282–285
  - capacity, 282
  - cards, 289–292
  - deep copying, 299–303
    - to class libraries, 302–303
  - definitions, 285–286
  - IComparable interface, 322–325
  - IComparer interface, 322–325
  - indexers, 285–289
    - implementation, 287–289
    - this keyword, 287
  - interfaces, 278
  - iterators, 294–299
    - definition, 294
    - implementation, 296–298
  - keyed, 292–294
  - lambda expressions, 426–429
    - application, 427–429
  - strongly typed, 278
  - use of, 278–285
- ColorDialog, 579–580**
  - application, 579
  - properties, 580
- colors, in GDI+ classes, 1118–1119**
- ColumnHeader, 478**
- Command windows, 175–176**
- CommandBuilder object, 938**
- command-line arguments, 126**
  - application, 145–146
  - execution, 146–147
  - Main() function, 145–147
- commands**
  - covert, 100–103
    - nothing else except spaces, 100
    - overflow checking, 101
  - interrupting loops, 88–89
  - SQL, in ADO.NET, 977–981
    - data modification, 982–983
    - direct execution, 980–981
    - single value retrieval, 981–982
    - string manipulation, 119
- comma-separated value (CSV), 797–801**
- comments, general, 33**
- comments, XML documentation, 1052–1059**
  - elements, 1053–1059
  - structural, 1056–1059
  - text formatting, 1054–1056
- common dialogs, 537–539**

- Common Language Runtime (CLR), 4**
- Common Object Request Broker Architecture (CORBA), 710**
- Common Type System (CTS), 4**
- comparison operators, 59–61**
  - overloading, 312
- comparisons, 303–325. See also type comparisons; value comparisons**
  - type, 303–308
    - boxing, 304–305
    - is operator, 306–308
    - unboxing, 304–305
  - value, 303, 309–325
    - IComparable interface, 320–325
    - IComparer interface, 320–325
    - operator overloading, 309–320
- compiling code, 5**
- composite controls. See user controls**
- compressed data, 802–805**
  - reading, 802–805
  - writing, 802–805
- conditional Boolean operators, 61–62**
- configuring exceptions, 182**
- connection zone, 697–701**
  - creation of, 698–701
  - properties, 699
- console applications, 13, 18–21**
  - CardLib, 274
  - Console.WriteLine(), 88
  - Console.WriteLine(), 35
    - mathematical operators, 50
  - Debug ⇨ Start Debugging menu, 20–21
  - execution, 21
  - initialization, 20
  - in Solution Explorer, 22
  - structure, 34–35
  - termination, 21
- Console.WriteLine(), 88**
- Console.WriteLine(), 35**
  - for loops, 88
  - mathematical operators, 50
- constraining types, 355–357**
- constructor execution sequence, 219–222**
  - System.Object, 219
- constructor initializers, 221**
- constructors, 190–191, 218–222**
  - C# class definitions, 218–222
    - initializers, 221
  - custom attributes, 1040
  - default, 190, 218
  - LINQ to XML, 994–998
    - element text with strings, 997–998
  - noncreatable, 218
  - nondefault, 190
    - class members, C# language, 249
    - static members, 192
- consumer objects, 938–939**
  - DataSet, 938
- containment, 199–200**
  - UML, 200
- content presenters, 1162**
- content syntax, 1162–1163**
  - content presenters, 1162
  - property element and, 1163
- controls, Visual Studio 2008, 524–535. See also user controls**
  - custom, 524
  - user, 524–535
    - debugging, 530–531
    - LabelTextbox, 526–530
- controls, Windows Forms Designer, 434–491**
  - anchoring, 435–437
    - properties, 437
  - button, 440–443
    - events, 441–443
    - properties, 440
  - CheckBox, 454, 456–457
    - events, 456
    - GroupBox, 457–458
    - properties, 456
  - CheckedListBox, 469–474
    - docking, 435–437
      - properties, 437
    - GroupBox, 457–461
      - application, 459–461
      - CheckBox, 457–458
      - RadioButton, 457–458
  - ImageList control, 478–479
  - label, 443–444
    - properties, 444
  - LinkLabel, 443–444
    - properties, 444
  - ListBox, 469–474
    - events, 472–474
    - methods, 471
    - properties, 470–471
  - ListView, 475–486
    - ColumnHeader, 478
      - events, 478
    - interfaces, 479
    - ListViewItem, 478
      - methods, 477
      - properties, 475–477
  - menustrip, 494–495
    - ToolStripMenuItem, 496–499
  - properties, 434–435
  - RadioButton, 454–455
    - events, 455
    - GroupBox, 457–458
    - properties, 455
  - RichTextBox, 461–469
    - events, 463–469
    - properties, 462–463
  - snapping, 435–437
    - snap-lines, 436–437
  - TabControl, 486–491
    - application, 488–489
    - event handlers, 490–491
    - properties, 487–488
  - TextBox, 444–454
    - application, 447–448
    - events, 446–454
    - properties, 445
- conventions, naming, 42–43**
  - camelCase, 42
  - explicit, code modification, 96
  - Hungarian notation, 42
  - PascalCase, 42
- conversions, 93–103, 326–328. See also explicit conversions; implicit conversions; type conversions**
  - boxing, 304–305

- explicit, 95–103
    - covert commands, 100–103
    - definition, 96
    - overflow checking, 97–99
    - short values into bytes, 96
  - implicit, 94–95
    - char types, 93–94
    - code, 94
    - short types, 95
    - storage values, 94
  - implicit conversions, 47, 94–95
  - operator overloading, 326–327
  - as operators, 327–328
  - type, 51, 93–103
    - application, 101–102
    - execution, 102–103
  - unboxing, 304–305
  - cookies, 647–648**
  - copying, deep v. shallow, 238**
    - collections, 299–303
  - CORBA. See Common Object Request Broker Architecture**
  - Corby, Karen, 1164**
  - covert commands, 100–103**
    - nothing else except spaces, 100
    - overflow checking, 101
  - CreateUserWizard, 659–660**
  - CSV. See comma-separated value**
  - CTS. See Common Type System**
  - custom activities, in WF, 1260–1266**
    - activity validation, 1263–1265
      - addition of, 1263–1264
    - design time enhancement, 1265–1266
  - custom attributes, 1038–1046**
    - AttributeUsage.AllowMultiple, 1043–1045
    - AttributeUsage.Inherited, 1046
    - BugFixAttribute, 1039
    - class usage, 1040–1041
    - constructors, 1040
    - properties, 1040
    - scope, 1042–1043
    - System.AttributeUsageAttribute, 1041–1042
    - target values, 1041–1042
  - custom controls, 524**
  - custom data types, 686**
    - binary serialization, 686
    - string serialization, 686
    - XML serialization, 686
  - custom exceptions, 373–375**
    - catch keyword, 177–178
    - finally keyword, 178
    - try keyword, 177–178
  - CustomValidator, 703–704**
- D**
- data access, history of, 934–935**
    - DBLib, 934
    - OCI, 934
    - ODBC, 934–935
  - data binding, 921–927**
    - WF, 1273–1277
      - bound property definitions, 1274–1277
  - data contracts, 1220, 1231**
  - DataAdapter object, 938**
  - DataReader object, 938, 940–948**
  - DataSet object, 938, 948–952, 965–973. See also rows, ADO.NET**
    - access options, 948–952
    - columns/rows/tables, 948–949
    - data fill, 948
    - multiple tables, 965–973
      - ADO.NET relationships, 965
      - navigation with, 966–973
  - DataTable object, 938–939**
  - DBLib, 934**
  - DCOM. See Distributed COM**
  - debugging, 155–176. See also break mode, debugging; nonbreak mode, debugging**
    - break mode, 166–176
      - Command windows, 175–176
      - Immediate windows, 175–176
    - call stack window, 176
    - configurations, 156
      - release, 156
    - directories, 156
    - nonbreak mode, 157–166
      - logging file creation, 157
      - outputting, 158–163
      - tracepoints, 163–165
    - stepping through code, 174–175
    - symbolic information, 156
    - user controls, 530–531
    - variable content monitoring, 171–174
    - VCE, 156–176
    - VS 2008, 156–176
  - Debug ⇌ Start Debugging, 20–21**
  - deep copying, 238**
    - collections, 299–303
  - deep copying, shallow v., 238**
  - default constructors, 190, 218**
    - object initializers, 400, 402
  - default statements, 76**
  - DelayActivity, 1251–1252**
  - delegates, 126, 150–153**
    - application, 151
    - definition, 152
    - event handlers, 376, 382–383
    - execution code, 151–152
    - generic, 344, 367
    - lambda expressions, 425–426
    - Main() function, 152
    - purpose, 150
    - snap-in, 153
  - delimited files, CSV, 797–801**
  - dependency properties, 1166–1167, 1201–1212**
  - deployment, of web applications, 759–772**
    - IIS, 628, 759–763
      - application pool creation, 761–762
      - configuration, 760
      - installation, 759–760
      - manager tool, 760
    - websites, 763–767
      - copying, 763–765
      - publishing, 765–767
    - Windows Installer, 767–772
      - installation, 769–772
      - properties, 769
      - setup program, 767–769

## **deployment, of Windows applications, 583–617**

- ClickOnce, 584–597
  - certifications, 586
  - manifest signing, 585–586
  - permission requirement definitions, 587–589
  - publish configuration options, 589–590
  - publish wizard, 590–597
- DLL hell, 583–584
- no-touch, 584
- overview, 583–584
- VS 2008, 597–598

## **desktop applications, in WPF, 1164**

## **destructors, 191**

- C# class definitions, 218–219

## **DetectURLs, 462**

## **diagnostics output, tracepoints v., 165–166**

## **dialogs, 537–581. See also OpenFileDialog; printing;**

### **SaveFileDialog**

- ColorDialog, 579–580
  - application, 579
  - properties, 580
- common, 537–539
- file, 540–559
  - OpenFileDialog, 540–553
  - SaveFileDialog, 553–559
- FolderBrowserDialog, 580–581
  - application, 581
  - properties, 581
- FontDialog, 577–579
  - application, 577
  - apply button, 578–579
  - properties, 578
- general applications, 539–540
- .NET Framework, 537–538
- PageSetupDialog, 568–570
  - as addition, 569–570
  - margins, 568
  - orientation, 569
  - paper, 568
  - printer properties, 569
- PrintDialog, 570–574
  - as addition, 571
  - options, 572
  - page ranges, 574
  - print selection, 572–574
  - properties, 574
- printing, 559–577
- PrintPreviewDialog, 575–576
  - PrintPreviewControl, 576
- public instance methods, 538
- ReadMe, 618
- user interface editor, 614–617
  - addition of, 615–617
  - default, 614–615
  - welcome, 618

## **Dictionary<K,V>, 350–351**

## **dimension control, 1175**

## **directory class, 780–781**

- properties, 781

## **DirectoryInfo class, 783–784**

- properties, 784

## **disposable objects, 194**

## **Distributed COM (DCOM), 711**

## **DLL hell, 583–584**

## **DNS. See Domain Name System**

## **do loops, 79–81**

- application, 79–80
- counter variables, 80
- structure, 79
- while v., 83

## **docking control, 435–437**

- properties, 437

## **DockPanel control, 1177–1179**

## **document type definitions (DTDs), 824**

## **documentation, XML, 1049–1072. See also comments, XML**

### **documentation**

- addition of, 1050–1052
  - with class diagrams, 1059–1063
  - viewing applications, 1050–1052
- comments, 1052–1059
  - elements, 1053–1058
- example application, 1066–1068
- file generation, 1063–1065
- processing for, 1068–1070
- tools, 1071–1072
  - NDoc, 1071
  - Sandcastle, 1072
  - with XSLT, 1070–1071

## **documents, XML, 819–828. See also XML document**

### **object model**

- attributes, 821
- declaration, 821–822
- DTDs, 824
- elements, 820–821
- invalid, 824
- LINQ to XML, 998–1002
  - loading, 998–1002
  - queries, 1009–1015
  - saving, 998–1002
- namespaces, 823
- nodes, 821, 831–847
  - Xpath, 839–847
- root elements, 822
- structure, 822–823
- validating parsers, 824
- VS 2008, 826–828
- well-formed, 824
- XML-DOM, 828–838
  - classes, 828–831
  - nodes, 831–839

## **Domain Name System (DNS), 1078–1080**

## **double parameters, for C#, 130, 135**

## **double-buffering, 1128, 1135–1137**

## **DoubleClick, 438**

## **downstream navigation, 973**

## **DragDrop, 438**

## **DragEnter, 438**

## **DragLeave, 438**

## **DragOver, 438**

## **DTDs. See document type definitions**

# E

## **EB. See Microsoft Expression Blend**

## **editor zone, 692–695**

- addition of, 694–695
- appearances, 693
- behavior, 693
- layout, 692–693

- elapsed events, 379, 384**
- elements, in arrays, 111**
- else blocks, 74**
- else statements, 71**
- Enabled, 435**
- endpoints, 1218–1220**
- enumerations, 93, 103–118**
  - application, 105–106
  - definition, 103–105
  - execution, 106–107
  - orientation types, 103, 106–107
  - structs, 109–110
  - typeof operators, 107
  - underlying types, 104
- Equals(object), 216**
- Equals(object,object), 216**
- error list window, 17, 23–24**
  - line number displays, 24
  - VCE, 24
  - VS 2008, 17
- errors, handling of, 155, 176–184**
  - configuring exceptions, 182
  - definition, 176
  - fatal, 155
  - listing exceptions, 182
  - logic/semantic, 155
  - namespaces, 177
  - SEH, 177–182
    - application, 178–180
      - catch keyword, 177
      - finally keyword, 177
      - notes, 183–184
      - throw exceptions, 180–182
      - try keyword, 177
    - VCE, 155, 176–184
    - VS 2008, 155, 176–184
- event(s), 202, 375–388**
  - anonymous methods, 388
  - anonymous users, 687
  - attached, 1173
  - button controls, 441–443
  - CardLib, 389–397
    - application, 389–397
    - expansion, 389
  - CheckBox control, 456
  - definition, 375–376, 379–384
    - application, 380–381
    - execution, 382–384
  - elapsed, 379, 384
  - handlers, 375–379
    - application, 377–378
    - ASP.NET, 637–641
    - button controls, 442–443
    - delegates, 376, 382–383
    - execution, 378–379
    - LabelTextbox, 529–530, 532–534
    - ListBox controls, 474
    - ListView control, 484–486
    - multipurpose, 384–387
    - PreviousPage creation, 640–641
    - results page creation, 639–640
    - return values, 388
    - sequencing, 376
    - subscription, 375
    - TabControl control, 490–491
    - TextBox controls, 448–454
    - toolbars, 505–508
    - ListView control, 478
    - PrintPage, 562–563, 565–567
      - modifications, 566–567
      - multiple pages, 565–567
    - RadioButton control, 455
    - RichTextBox control, 463–469
    - routed, 1168–1173
    - TextBox controls, 446–454
    - Windows Forms Designer controls, 437–439
- event handlers, 375–379**
  - application, 377–378
  - ASP.NET, 637–641
  - button controls, 442–443
  - delegates, 376, 382–383
  - execution, 378–379
  - LabelTextbox, 529–530, 532–534
  - ListBox controls, 474
  - ListView control, 484–486
  - multipurpose, 384–387
    - application, 384–387
    - execution, 387
  - PreviousPage creation, 640–641
  - results page creation, 639–640
  - return values, 388
  - sequencing, 376
  - subscription, 375
  - TabControl control, 490–491
  - TextBox controls, 448–454
  - toolbars, 505–508
- executable hosting, 1221**
- explicit conversions, 95–103**
  - code modification, 96
  - covert commands, 100–103
    - nothing else except spaces, 100
    - overflow checking, 101
  - definition, 96
  - overflow checking, 97–99
    - context, 97
    - covert commands, 101
    - options, 99
    - Solution Explorer, 98
  - short values into bytes, 96
- expression trees, 425–426**
- expressions, 46–56. See also operators**
  - expression trees, 425–426
    - lambda, 418–429
      - anonymous methods, 418–423
      - collections, 426–429
      - delegates, 425–426
      - LINQ, 418–429
      - MSIL code, 419
      - parameters, 423
      - simple, 420–423
      - statement bodies, 424–425
  - operators, 46–53
    - assignment, 40, 51–52
    - binary, 46–48, 52
    - Boolean values, 46
    - mathematical, 46–51
    - precedence, 52–53
    - ternary, 46
    - unary, 46–48

## **extender controls, 755–757**

### **Extensible Application Markup Language (XAML), 1145–1147**

- Microsoft EB, 1145–1146
- syntax, 1161–1163
  - attribute, 1161–1162
  - content, 1162–1163
  - markup extensions, 1163
  - object element, 1161
  - property element, 1162

### **Extensible Markup Language (XML), 819–847. See also Asynchronous Javascript and XML; documentation, XML; documents, XML; nodes; XML document object model**

- ADO.NET, 936, 973–977
    - DataSet supports, 973
  - AJAX, 739–757
    - client script extensions, 742
    - controls, 741, 746–749
    - overview, 739–741
    - update panels, 741–746
    - web services, 749–754
  - documentation, 1049–1072
    - addition of, 1050–1052
    - comments, 1052–1059
    - example application, 1066–1068
    - file generation, 1063–1065
    - processing for, 1068–1070
    - tools, 1071–1072
    - with XSLT, 1070–1071
  - documents, 819–828
    - attributes, 821
    - declaration, 821–822
    - DTDs, 824
    - elements, 820–821
      - invalid, 824
    - namespaces, 823
    - nodes, 821, 831–847
    - root elements, 822
    - structure, 822–823
    - validating parsers, 824
    - VS 2008, 826–828
    - well-formed, 824
    - XML-DOM, 828–838
  - LINQ to XML, 850, 993–1015
    - constructors, 994–998
    - documents, 998–1002
    - fragments, 1002–1004
    - from LINQ to SQL, 1004–1008
    - Northwind Sample database, 1008–1009
  - schemas, 824–828
    - XRD, 824
    - XSD, 825–828
  - serialization, 686
  - SOAP 711
  - XAML, 1145–1147
- ### **extension methods, 412–417**
- application, 413–415
  - definition parameters, 412–413
  - execution, 415–417
  - LINQ, 417, 855
  - ReverseString(), 415–417
  - ToStringReversed(), 415–416
  - ToTitleCase(), 415, 417

## **F**

### **fatal errors, 155**

### **fault contracts, 1220, 1233**

### **fields, OOP techniques, 188–189**

- private, 188

### **file class, 780–781**

- properties, 781

### **file dialogs, 540–559**

- OpenFileDialog, 540–553
  - addition of, 550–553
  - directory specification, 541–542
  - filter setting, 542–543
  - help, 544
  - properties, 545
  - results, 544
  - text editor application, 545–550
  - title, 541
  - validation, 543–544
- SaveFileDialog, 553–559
  - as addition, 555–558
  - extensions, 554
  - overwriting files, 554
  - properties, 555
  - title, 554
  - validation, 554

### **file system data, 777–817. See also streams**

- serialized objects, 805–810
- streams, 778–805
  - input/output classes, 778–805
  - serial device, 778
- structure monitoring, 810–817
- properties, 811–812

### **file systems editor, 607–610**

- file addition, 609–610
- folder additions, 608
- properties, 608–609

### **File Transfer Protocol (FTP), 710**

- networking, 1078

### **file types editor, 611–612**

- action creation, 611
- extension, 612
- properties, 611

### **FileInfo class, 781–783**

- properties, 782–783

### **FileStream object, 784–791**

- data, 787–791
  - reading, 787–789
  - writing, 789–791
- file position, 786

### **finally, as keyword, 178**

### **firewalls, 716**

- SOAP, 716

### **First(), 886–887**

### **FirstOrDefault(), 886–887**

### **float values, 102**

### **flow control, 59–90. See also Boolean logic; looping**

- Boolean logic, 59–69
  - goto statement, 69–70
  - operators, 46, 61–69
  - values, 60–61
- branching, 59, 70–78
  - if statement, 71–75

- switch statement, 75–78
- ternary operators, 46, 70–71
- looping, 59, 78–90
  - for, 83–88
  - do loops, 79–81
  - infinite, 89–90
  - interrupting, 88–89
  - while, 81–83
- FolderBrowserDialog, 580–581**
  - application, 581
  - properties, 581
- font class, GDI+ classes, 1124–1128**
- FontDialog, 577–579**
  - application, 577
  - apply button, 578–579
  - properties, 578
- for loops, 83–88**
  - application, 85
  - Console.WriteLine(), 88
  - counter variables, 84
  - definition, 83–84
  - execution, 86–88
    - Mandelbrot sets, 86
  - format, 84
  - iteration, 87–88
  - structure, 84
  - switch statements, 88
- foreach loops, 114–117**
  - jagged arrays, 117
  - LINQ, 854
  - multidimensional arrays, 116
  - read-only access, 114
- ForeColor, 435**
- Form1.cs file, 27–28**
- Form1.Designer code, 27**
- from clause, 853**
- FTP. See File Transfer protocol**
- functional programming, 186**
- functions, for C#, 125–154**
  - code, 125, 127
    - collapsing, 125
    - multipurpose, 126
    - reusable, 125
  - command-line arguments, 126, 145–147
    - application, 145–146
    - execution, 146–147
  - data exchange applications, 130–131
  - definition, 126–138
    - entry point, 127
    - return values, 128–130
  - delegates, 126, 150–153
    - application, 151
    - definition, 152
    - execution code, 151–152
    - purpose, 150
    - snap-in, 153
  - execution code, 131–132
  - Main(), 144–147, 152
    - application, 145–146
    - execution, 146–147
  - overloading, 126, 148–150
  - parameters, 130–138
    - arrays, 133–135
    - double, 130, 135
    - global data v., 143–144
    - matching, 132–133
    - out, 137–138
    - reference, 135–136
    - value, 135–136
  - PascalCase, 42
  - purpose, 125
  - signatures, 126, 149
  - struct, 147–148
    - static modifiers, 148
  - toolbars, 501–512
  - use of, 126–130
  - variable scope, 138–144
    - branching, 141
    - definition, 138–141
    - global, 140
    - local, 140
    - looping, 141
    - operations, 139
    - out of scope, 139
    - use of, 138–141

## G

- GAC. See Global Assembly Cache**
- GDI+ classes. See Graphics Device Interface classes**
- general comments. See comments, general**
- generic(s), 331–367**
  - CardLib, 351
  - classes, 332, 352–364
    - constraining types, 355–357
    - default keyword, 354–355
    - definition, 352–364
    - inheritance, 361–362
    - operators, 363–364
    - reflection of, 353
    - structs, 364
    - unbounded types, 355
  - definition, 331–333, 351–367
  - delegates, 344, 367
    - definition, 367
  - interfaces, 365
  - methods, 365–367
  - nullable types, 333–340
    - operators, 334–340
  - sorting, 344–349
  - System.Collections.Generic namespace, 333–351
    - Dictionary<K,V>, 350–351
    - List<T>, 341–344
- generic classes, 332, 352–364**
  - constraining types, 355–357
  - default keyword, 354–355
  - definition, 352–364
    - application, 355–357
    - inheritance, 361–362
  - operators, 363–364
    - overrides, 363
  - reflection of, 353
  - structs, 364
  - unbounded types, 355
- generic delegates, 344, 367**
- generic methods, 365–367**
- generic operators, 363–364**

**GetHashCode(), 217, 411**  
**GetType(), 217**  
**Global Assembly Cache (GAC), 5**  
**global namespaces, 53**  
**global variables, 140**  
    hidden, 141  
    parameter functions v., 143–144  
    return values v., 143–144  
**GotFocus, 438**  
**goto statements, 69–70**  
    interrupting loops, 89  
    spaghetti code, 69  
**graphics class, 1108–1109**  
**Graphics Device Interface (GDI+) classes, 1107–1140**  
    advanced capabilities, 1138–1140  
    clipping, 1138–1139  
        System.Drawing.Drawing2D, 1139–1140  
        System.Drawing.Imaging, 1140  
    colors, 1118–1119  
    coordinate systems, 1110–1118  
        GraphicsPaths, 1114–1116  
        point, 1113  
        rectangle, 1114  
        regions, 1116–1118  
        size, 1114  
    graphics class, 1108–1109  
    image creation, 1128–1131  
        double-buffering, 1128, 1135–1137  
    line drawing, 1119–1121  
        pen class, 1119–1121  
    object disposal, 1109–1110  
    overview, 1108  
    shape drawing, 1122–1124  
        brush class, 1122–1124, 1131–1133  
    text drawing, 1124–1128  
        font class, 1124–1128  
**GraphicsPaths, in GDI+ classes, 1114–1116**  
**grid control, 1179–1181**  
**GridView control, 663–669**  
    configurations, 668–669  
**group queries, 881–883, 914–917**  
**GroupBox control, 457–461**  
    application, 459–461  
    CheckBox control, 457–458  
    RadioButton control, 457–458

## H

**Hejlsberg, Anders, 399, 1070**  
**hosting options, 1221**  
    executable, 1221  
    WAS, 1221  
    web server, 1221  
    Windows service, 1221  
**HTML code, 628–637**  
    ASP.NET, 628–637  
    server controls, 636–637  
**HTTP. See Hypertext Transfer protocol**  
**Hungarian notation, 42**  
**Hypertext Transfer Protocol (HTTP), 710**  
    addresses, 1218  
    ASP.NET, 628  
    networking, 1078  
    WCF, 1217

**I**  
**ICollection, 278, 285, 292**  
**Icomparable interface, 320–325**  
    collection sorting, 322–325  
    execution, 325  
**Icomparer interface, 320–325**  
    collection sorting, 322–325  
    execution, 325  
    .NET Framework, 321  
**IDE. See Integrated Development Environment**  
**IDictionary, 278, 292–294**  
**IEnumerable, 278, 283, 285, 292**  
**if statements, 71–75**  
    application, 72–74  
    conditions, 74–75  
    else blocks, 74  
    else statements w., 71  
**IIS. See Internet Information Services**  
**IList, 278, 285, 287**  
**image creation, in GDI+ classes, 1128–1131**  
    double buffering, 1128  
**ImageList control, 478–479**  
**implicit conversions, 47, 94–95**  
    char types, 93–94  
    code, 94  
    short types, 95  
    storage values, 94  
**indentations, 32**  
**indexers, 285–289**  
    implementation, 287–289  
    this keyword, 287  
**infinite loops, 89–90**  
    break statements, 90  
    Windows Task Manager, 90  
**inheritance, 195–197**  
    classes, 195–197  
        abstract, 196–197  
        parent, 195  
        sealed, 197  
        virtual, 196  
    generic classes, 361–362  
    member return types, 195  
    purpose, 195  
    UML, 195  
**initializers, 400–406**  
    collection, 402–406  
        application, 403–404  
        execution, 405–406  
    object, 400–402  
        default constructor, 400, 402  
        nested, 402  
        properties, 400–401  
**installation. See deployment, of Windows applications; Windows Installer**  
**integer types, 37**  
    unsigned, 37  
**Integrated Development Environment (IDE), 9**  
    auto-completion, 123  
    VS 2008  
        error list, 17  
        layout, 16  
        menus, 16  
        preference settings, 14–16  
        properties window, 17

- Solution Explorer, 16–17
- toolbars, 16
- writing programs, 14–17

**Intellisense**

- VCE, 149
- VS 2008, 149

**interface polymorphism, 198–199****interfaces, 193–194, 257–261**

- class definitions, 212–215, 233–235
  - application, 213–214
  - execution, 214–215
- class members, 257–261
  - accessors, 261
  - explicit, 260–261
  - implementation, 257–261
  - implicit, 260–261
- collections, 278
- GDI+ classes, 1107–1140
- generics, 365
- IComparable, 320–325
- IComparer, 320–325
- ListView control, 479
- lollipop syntax, 193
- MDI, 493, 512–524
- multiple, 194
- OCI, 934
- polymorphism, 198–199
- SDI, 493, 512–514
- System.Collections.Generic namespace, 333–351
  - Dictionary<K,V>, 350–351
  - List<T>, 341–344

**Internet Information Services (IIS), 628, 759–763**

- application pool creation, 761–762
- configuration, 760
- installation, 759–760
- manager tool, 760

**Internet Protocol (IP) addresses, 1078–1080****interrupting loops, 88–89**

- commands, 88–89
- goto statements, 89

**IP addresses. See Internet Protocol addresses****is operator, 306–308**

- application, 306–308
- execution, 308

**iterators, 294–299**

- definition, 294
- implementation, 296–298

**J****jagged arrays, 116–117**

- foreach loops, 117

**JavaScript, 32, 701–707. See also Asynchronous****Javascript and XML**

- AJAX, 739–757
  - client script extensions, 742
  - controls, 741, 746–749
  - overview, 739–741
  - update panels, 741–746
  - web services, 749–754
- controls, 706–707
- CustomValidator, 703–704
- function definition, 702
- .NET Framework compatibility, 4

- objects, 703–705
- runtime, 701
- script element, 701
- statements, 702–703
- variable declaration, 702
- window opening, 705–706

**JIT compilers. See Just-in-Time compilers****joins, 891–892****JScript language, .NET Framework compatibility, 4****Just-in-Time (JIT) compilers, 5, 7****K****KeyDown, 438****keyed collections, 292–294****KeyPress, 438****KeyUp, 438****keywords**

- catch, 177–178
- default, 354–355
  - generic classes, 354–355
- finally, 178
- partial, 635
- this, 256–257
- try, 177–178

**L****label controls, 443–444**

- properties, 444

**LabelTextbox control, 526–530**

- event handler addition, 529–530, 532–534
- extension, 531–535
- property addition, 528–529, 532

**lambda expressions, 418–429**

- anonymous methods, 418–423
- collections, 426–429
- delegates, 425–426
- expression trees, 425–426
- LINQ, 418–429, 856–858
- MSIL code, 419
- parameters, 423
- simple, 420–423
- statement bodies, 424–425

**Language Integrated Query (LINQ), 9, 417–429, 849–892.****See also ActiveX Data Objects.NET**

- aggregate operators, 865–868
  - numeric, 865–868
  - properties, 865
- Boolean values, 877–878
  - All(), 877–878
  - Any(), 877–878
- complex objects, 868–872
- First(), 886–887
- first query, 850–854
  - from clause, 853
  - deferred execution, 854
  - foreach loop, 854
  - select clause, 854
  - variable declaration, 852–853
  - where clause, 853
- FirstOrDefault(), 886–887
- group queries, 881–883

# Language Integrated Query (LINQ) (continued)

---

## Language Integrated Query (LINQ) (continued)

- joins, 891–892
- lambda expressions, 418–429, 856–858
  - anonymous methods, 418–423
- large data sets, 862–864
- LINQ over Dataset, 985–989
  - applications, 985
- LINQ to SQL, 850, 895–929
  - advanced queries, 914–917
  - bound data updating, 928–929
  - data binding, 921–927
  - drilling down with, 911–914
  - first query, 898–907
  - generated displays of, 917–921
  - grouping queries, 914–917
  - ordering queries, 914–917
  - ORM, 896
  - relationship navigation in, 907–911
  - XML from, 1004–1008
- LINQ to XML, 850, 993–1015
  - constructors, 994–998
  - documents, 998–1002
  - fragments, 1002–1004
  - from LINQ to SQL, 1004–1008
  - Northwind Sample database, 1008–1009
- methods, 855–856
  - extension, 855
  - syntax, 855–856
- orderby clause, 860
- ordering in, 860–862
  - method syntax, 881
  - multiple levels, 879–881
- projection, 872–875
  - method syntax, 875
  - object creation, 872–874
- query results, 858–859
- resources, 892
- select distinct query, 875–876
- set operators, 887–891
- Skip(), 883–885
- Take(), 883–885
- 3.0 enhancements, C# language, 417
- varieties
  - LINQ providers, 850
  - LINQ to objects, 850
  - LINQ to SQL, 850
  - LINQ to XML, 850
- languages. See C# language; C++ language; COBOL language; Extensible Markup Language; JavaScript language; 3.0 enhancements, C# language; Universal Modeling Language**
- launch condition editor, 612–613**
- layout editor, 692–693**
- license agreements, 619**
- line drawing, in GDI+ class, 1119–1121**
- line number displays, 24**
- linking, .NET Framework, 7–8**
- LinkLabel control, 443–444**
  - properties, 444
- LINQ. See Language Integrated Query (LINQ)**
- LINQ to Objects, 850**
- LINQ to SQL, 850, 895–929**
  - advanced queries, 914–917
  - bound data updating, 928–929
  - data binding, 921–927
  - drilling down with, 911–914
  - first query, 898–907
    - O/R Designer, 904–906, 909–910
  - generated displays of, 917–921
  - grouping queries, 914–917
  - ordering queries, 914–917
  - ORM, 896
  - relationship navigation in, 907–911
  - XML from, 1004–1008
- LINQ to XML, 850, 993–1015**
  - constructors, 994–998
    - element text with strings, 997–998
  - documents, 998–1002
    - loading, 998–1002
    - queries, 1009–1015
    - saving, 998–1002
  - fragments, 1002–1004
  - from LINQ to SQL, 1004–1008
  - Northwind Sample database, 1008–1009
- Linux operating system, 3**
- ListBox control, 469–474**
  - events, 472–474
    - handlers, 474
  - methods, 471
  - properties, 470–471
- listing exceptions, 182**
- List T, 341–344**
  - application, 343–344
  - methods, 342
  - properties, 342
  - searching/sorting, 345–347
- ListView control, 475–486**
  - ColumnHeader, 478
  - events, 478
    - handlers, 484–486
  - interfaces, 479
  - ListViewItem, 478
  - methods, 477
  - properties, 475–477
- ListViewItem, 478**
- literal values, 43–45**
  - string, 44–45
  - suffixes, 43
- local variables, 140**
- logic errors, 155**
- login page, creation, 658–659**
- lollipop syntax, 193**
- looping, 48, 59, 78–90**
  - for, 83–88
    - application, 85
    - Console.Write(), 88
    - counter variables, 84
    - definition, 83–84
    - execution, 86–88
    - format, 84
    - iteration, 87–88
    - structure, 84
    - switch statements, 88
  - do, 79–81
    - application, 79–80
    - counter variables, 80
    - structure, 79
    - while v., 83
  - foreach, 114–117
    - jagged arrays, 117

- multidimensional arrays, 116
- infinite, 89–90
  - break statements, 90
  - Windows Task Manager, 90
- interrupting, 88–89
  - commands, 88–89
  - goto statements, 89
  - variable scope, 141
- while, 81–83
  - application, 82
  - Boolean test, 81
  - do v., 83
  - execution, 83
  - false evaluation, 81
  - user validation, 83
- XML-DOM nodes, 831–833

**LostFocus, 438**

## M

**MAC OS operating system, 3**

**Main() function, 144–147**

- application, 145–146
- command-line arguments, 145–147
- delegate code, 152
- execution, 146–147

**Mandelbrot sets, 86–87**

**manifest signing, 585–586**

**margin control, 568, 1175**

**master pages, 671–678**

- application, 675–676
- creation, 673–676
- use of, 677–678

**mathematical operators, 46–51**

- Console.WriteLine(), 50
- conversions
  - implicit, 47
  - type, 51
- looping, 48
- manipulation of, 49–51

**MaxLength, 445**

**MDI. See Multiple Document Interface**

**member return types, 195**

**members. See class members, C# language**

**membership provider, 652**

**MemberwiseClone(), 217**

**menus, 494–501**

- functionality additions, 499–501
- manual creation, 495–498
- menustrip control, 495–499
  - ToolStripMenuitem, 496–499
- merging, 519–522
- text editors, 519–522
- Visual Studio 2008, 16
- VS 2008, 16, 494–501

**menustrip control, 494–495**

- ToolStripMenuitem, 496–499

**merging menus, 519–522**

**message contracts, 1220, 1233**

**message patterns, WCF, 1220–1221**

**Message Transmission Optimization Mechanism (MTOM), 1216**

**methods, for C# language**

- addition of, 250–251

- anonymous, 388
  - lambda expressions, 418–423
- definition, 242–244
- extension, 412–417
  - application, 413–415
  - definition parameters, 412–413
  - execution, 415–417
  - LINQ, 417
  - ReverseString(), 415–417
  - ToStringReversed(), 415–416
  - ToTitleCase(), 415, 417
- hidden, 254–256
- ListBox controls, 471
- Listview control, 477
- override, 243, 254–256
- partial, 264–265
- public instance, 538

**methods, in OOP techniques, 189–190**

**methods, web services architecture, 714–716**

**Microsoft Expression Blend (EB), 1145–1146**

**Microsoft Installer (MSI) database, 599. See also SimpleEditor**

**Microsoft Intermediate Language (MSIL) code, 5, 7**

- assemblies, 5
- lambda expressions, 419

**Mono, 3**

**MouseDown, 438**

**MouseMove, 438**

**MouseUp, 438**

**MSI database. See Microsoft Installer database**

**MSIL. See Microsoft Intermediate Language code**

**MSMQ technology, WCF, 1217**

**MTOM. See Message Transmission Optimization Mechanism**

**multidimensional arrays, 115–116**

- foreach loops, 116

**Multiline, 445**

**Multiple Document Interface (MDI), 493, 512–524**

- applications, 514–516
  - child, 514
  - containers, 514
  - execution, 515–516
  - text editors, 517–519
  - merging menus, 519–522
  - properties, 519
  - tracking, 522–524

**multipurpose code, 126**

**multipurpose event handlers, 384–387**

- application, 384–387
- execution, 387

**multitier applications, ADO.NET, 936**

- business tier, 936
- data tier, 936
- presentation tier, 936

## N

**named pipe protocol, 1217–1218**

- addresses, 1218

**namespace statements, 35**

**namespaces, 53–56**

- error handling, 177
- global, 53
- levels, 54–56
- System.Data, 939–940

## namespaces (continued)

---

### namespaces (continued)

- native providers, 940
- ODBC .NET data provider, 940
- OLE DB .NET data provider, 939
- Oracle .NET Data provider, 939
- SQL Server .NET data provider, 939
- XML documents, 823
- URI, 823

### naming conventions. *See* conventions, naming

### naming, of variables, 41–43

- conventions, 42–43
- camelCase, 42
- Hungarian notation, 42
- PascalCase, 42
- rules, 41

### native code, 5

### NDoc, 1071

### ndoc.sourceforge.net, 1071

### nested object initializers, 402

### nesting types, 257

### .NET Framework, 3–8, 185–206. *See also* ActiveX Data

**Objects.NET; ASP.NET; attributes; documentation, XML; Extensible Markup Language; printing; streams**

- ADO.NET, 9, 933–989
  - database updating, 952–965
  - DataReader, 940–948
  - DataSet, 948–952
  - definition, 934
  - design goals, 935–936
  - history, 935
  - LINQ over DataSet, 985–989
  - nomenclature, 934
  - objects, 937–939
  - SQL support, 977–985
  - XML, 973–977
- ASP.NET, 627–669
  - authentication/authorization, 652–660
  - event handlers, 637–641
  - HTML code, 628–637
  - HTTP, 628
  - input validation, 642–646
  - overview, 628
  - runtime, 628
  - server controls, 636–637
  - SQL server database, 660–669
  - state management, 646–651
  - user input display, 638–639
  - web site creation, 629–636
- attributes, 1019–1046
  - assembly, 1020–1023
  - built-in, 1026–1027
  - custom, 1038–1046
  - definition, 1019–1023
  - reflection, 1023–1026
  - System.Diagnostics.ConditionalAttribute, 1027–1029
  - System.ObsoleteAttribute, 1029–1031
  - System.Reflection.AssemblyDelaySignAttribute, 1034–1038
  - System.SerializableAttribute, 1031–1034

### C# language compatibility, 4

### C++ language compatibility, 4

### class definitions, 193–197, 210, 233–235

- abstract, 196–197, 210, 233–235
- constructors, 218–222
- copying, deep v. shallow, 238
- destructors, 218–219

### interfaces, 193–194, 212–215, 233–235

- internal, 210
- library projects, 229–233, 269–274
- modifiers, 212
- objects, 187
- parent, 195
- partial, 262–263
- sealed, 197, 210
- virtual, 196

### class members, 241–276

- application, 247–248
- class diagrams, 249–252
- definitions, 241–242
- execution, 248–249
- fields, 242
- internal, 241
- methods, 242–244, 250–251, 254–256
- nondefault constructors, 249
- private, 241
- properties, 244–247
- protected, 241
- public, 241
- refactoring, 252–253
- this keyword, 256–257

### COBOL language compatibility with, 4

- destructors, 191
- dialogs, 537–538
- file system data, 777–817
  - serialized objects, 805–810
  - streams, 778–805
  - structure monitoring, 810–817
- generics, 332–351

### classes, 332

### IComparer interface, 321

### JavaScript language compatibility, 4

- library of, 4
  - basic types in, 4
  - CLR in, 4
  - CTS in, 4
  - OOP techniques in, 4

### Linux operating system, 3

### Mono operating system, 3

### no-touch deployment, 584

### OOP techniques, 4, 185–206, 223–229

- class diagrams, 227–229
- Class View window, 223–225
- components, 187–191
- custom exceptions, 373–375
- definition, 186–192
- events, 202, 375–388
- functional, 186
- inheritance, 195–197
- instance class members, 191–192
- Object Browser, 225–226
- operator and the global namespace qualifier, 371–373
- operator overloading, 201–202
- polymorphism, 197–199
- reference v. value types, 202
- static members, 191–192
- structure, 186
- UML, 187, 192–193, 195, 200
- Windows applications, 203–206

### PascalCase, 243

### PDAs, 3–4

### printing, 559–577

- architecture, 560–561
  - multiple pages, 565–567
  - PrintDialog, 570–574
  - PrintDocument, 560–565
  - PrintPage event, 562–563, 565–567
  - PrintPreviewDialog, 575–576
  - sequence, 561–562
  - System.Object, 211, 215–218
    - constructor execution sequence, 219
    - methods, 216–217
  - Two's complement, 64
  - VCE, 4
  - VS 2008, 4
  - web services, 717–720
    - alternative client protocols, 719–720
    - clients, 719
    - properties, 717
    - SoapHttpClientProtocol, 719
    - WebMethod attribute, 718
    - WebService attribute, 718
    - WebServiceBinding attribute, 719
  - writing applications with, 4–8
    - assemblies, 5
    - compiling code, 5
    - GAC, 5
    - garbage collection, 6
    - JIT compilers, 5, 7
    - linking, 7–8
    - managed code, 6
    - MSIL code, 5, 7
    - native code, 5
  - XML, 819–847
    - AJAX, 739–757
    - documentation, 1049–1072
    - documents, 819–828
    - LINQ to XML, 850
    - schemas, 824–828
    - serialization, 686
    - SOAP 711
  - networking, 1075–1104**
    - through broadcast, 1075
    - DNS, 1078–1080
    - FTP, 1078
    - HTTP, 1078
    - IP addresses, 1078–1080
    - layers, 1076–1078
    - overview, 1075–1078
    - programming options, 1083–1084
    - SMTP, 1078
    - TCP, 1078, 1081–1083
      - application protocols, 1081–1083
    - TcpClient, 1096–1104
      - client creation, 1099–1104
      - controls, 1100
    - TcpListener, 1096–1104
      - server creation, 1096–1099
    - UDP, 1078, 1081–1083
      - application protocols, 1081–1083
    - URI, 1080–1081
    - WebClient, 1084–1086
      - properties, 1084–1086
    - WebRequest, 1084–1095
      - controls, 1088
    - WebResponse, 1084–1095
      - controls, 1088
  - nodes, 821, 831–847**
    - XML-DOM, 831–839
      - creation of, 834–837
      - deletion of, 838–839
      - looping, 831–833
      - properties, 834–835
      - value changes, 833–836
    - XPath, 839–847
      - common operations, 840
      - queries, 847
  - nonbreak mode, debugging, 157–166**
    - logging file creation, 157
    - outputting, 158–163
      - application, 158–161
      - execution code, 161–162
      - indices, 162
    - tracepoints, 163–165
  - nonconstant variables, 136**
  - nondefault constructors, 190**
    - class members, C# language, 249
  - Northwind Sample database, 898–907. See also LINQ to SQL**
    - LINQ to XML, 1008–1009
  - nothing else except spaces, 100**
  - no-touch deployment, 584**
  - null coalescing operator, 335–340**
  - nullable types, 333–340**
    - operators, 334–340
      - applications, 336–339
  - numeric aggregate operators, 865–868**
- ## O
- Object(), 216**
  - object(s), 187, 190–191. See also collections; consumer objects; namespace statements; namespaces; provider objects; type comparisons; value comparisons**
    - ADO.NET, 937–939
      - consumer, 938–939
      - provider, 937–938
    - classes of, 187
    - collections, 200–201, 277–303
      - arrays v., 200, 277–285
      - capacity, 282
      - cards, 289–292
      - deep copying, 299–303
      - definitions, 285–286
      - indexers, 285–289
      - interfaces, 278
      - iterators, 294–299
      - keyed, 292–294
      - strongly typed, 278
      - use, 278–285
    - comparisons, 303–324
      - type, 303–308
      - value, 303, 309–325
    - complex, in LINQ, 868–872
    - constructors, 190–191
      - default, 190
      - nondefault, 190
    - containment, 199–200
    - destructors, 191
    - disposable, 194
    - event consumption, 202
    - FileStream, 784–791

## **object(s) (continued)**

- data, 787–791
  - file position, 786
  - GDI+, disposal of, 1109–1110
  - instantiate, 187
  - JavaScript, 703–705
  - life cycle, 190–191
  - methods, 189–190
  - reference types, 237
  - serialized, 805–810
  - StreamReader
    - data reading, 796
    - delimited files, 797–801
  - StreamWriter, 791–794
  - structs, 202
  - UML syntax, 187
- Object Browser, 225–226**
- object element syntax, 1161**
- object initializers, 400–402**
- default constructor, 400, 402
  - nested, 402
  - properties, 400–401
- Object Management Group (OMG), 710**
- object-oriented programming (OOP) techniques, 4, 185–206, 223–229, 371–397. See also collections; event(s); object(s)**
- class diagrams, 227–229
  - Class View window, 223–225
    - symbols, 224
  - collections, 200–201, 277–303
    - arrays v., 200, 277–285
    - capacity, 282
    - cards, 289–292
    - deep copying, 299–303
    - definitions, 285–286
    - indexers, 285–289
    - interfaces, 278
    - iterators, 294–299
    - keyed, 292–294
    - strongly typed, 278
    - use of, 278–285
  - comparisons, 303–325
  - components, 187–191
    - accessibility of, 188
    - fields, 188–189
    - methods, 189–190
    - objects, 187, 190–191
    - properties, 188–189
  - custom exceptions, 373–375
    - base classes, 373
    - CardLib, 374–375
    - definitions, 373
  - definition, 186–192
  - events, 202, 375–388
    - anonymous methods, 388
    - CardLib, 389–397
    - definition, 375–376, 379–384
    - elapsed, 379, 384
    - handlers, 375–379
  - functional, 186
  - inheritance, 195–197
    - classes, 195–197
    - member return types, 195
    - purpose, 195
  - instance class members, 191–192
  - interfaces, 193–194
    - lollipop syntax, 193
    - multiple, 194
    - polymorphism, 198–199
  - Object Browser, 225–226
  - operator and the global namespace qualifier, 371–373
  - operator overloading, 201–202
  - polymorphism, 197–199
    - interface, 198–199
  - reference v. value types, 202
  - static members, 191–192
    - constructors, 192
  - structure, 186
  - UML, 187, 192–193, 195, 200
  - VCE, 223–229
  - VS 2008, 223–229
  - Windows applications, 203–206
- object-relational mapping (ORM), 896**
- objects, System.Data namespace, 939–940**
- OCI. See Oracle Call Interface**
- ODBC. See Open Database Connectivity**
- ODBC .NET data provider, 940**
- OLE DB .NET data provider, 939**
- OMG. See Object Management Group**
- OOP techniques. See object-oriented programming techniques**
- Open Database Connectivity (ODBC), 934–935**
- OpenFileDialog, 540–553**
- as addition, 550–553
  - directory specification, 541–542
  - filter setting, 542–543
  - help, 544
  - properties, 545
  - results, 544
  - text editor application, 545–550
  - title, 541
  - validation, 543–544
- operation contracts, 1220, 1232–1233**
- operator and the global namespace qualifier, 371–373**
- ?? operator. See null coalescing operator**
- operator overloading, 201–202**
- binary, 312
  - comparison, 312
  - conversions, 326–327
  - functions, 126
  - OOP techniques, 201–202
  - unary, 312
  - value comparisons, 309–320
- operators, 46–53, 334–340. See also mathematical operators**
- as, 327–328
  - aggregate, 865–868
    - numeric, 865–868
    - properties, 865
  - assignment, 40, 51–52
    - Boolean, 66
  - binary, 46–48, 52
  - Boolean, 46, 61–69
    - application of, 67
    - assignment, 66
    - bitwise, 62–66
    - conditional, 61–62
    - precedence updates, 68–69
  - comparison, 59–61
    - overloading, 312

- generic, 363–364
- is, 306–308
- mathematical, 46–51
  - Console.WriteLine(), 50
  - implicit conversion, 47
  - looping, 48
  - manipulation of, 49–51
  - null coalescing, 335–340
  - nullable types, 334–340
    - applications, 336–339
  - partitioning, 884
  - precedence, 52–53
  - set, 887–891
  - ternary, 46, 70–71
  - typeof, 107
  - unary, 46–48
    - overloading, 312
- O/R Designer, 904–906, 909–910**
  - pluralization, 906
- Oracle Call Interface (OCI), 934**
- Oracle .NET Data provider, 939**
- orderby clause, 860**
- ordering, in LINQ, 860–862, 879–881**
  - LINQ to SQL, 914–917
  - method syntax, 881
  - multiple levels, 879–881
- ordering queries, 914–917**
- orientation types, 103, 106–107**
  - byte conversions, 107
  - enumerations, 109–110
- ORM. See object-relational mapping**
- out parameters, 137–138**
- overflow, 97–99**
  - checking context, 97
  - options, 99
  - Solution Explorer, 98
- overflow checking context, 97**
- overloading**
  - functions, 126
  - operator, 201–202, 309–320
    - conversions, 326–327
- override methods, 243, 254–256**
  - generic classes, 363
    - nonconstant variables, 136
    - value, 135–136
- parent classes, 195**
- partial class definitions, 262–263**
- partial keywords, 635**
- partial methods, 264–265**
- partitioning operators, 884. See also Skip(); Take()**
- PascalCase, 42**
  - C# functions, 42
  - public class members, 243
- PasswordChar, 445**
- path names, 784**
  - relative, 784
- PDAs. See personal digital assistants**
- pen class, in GDI+ class, 1119–1121**
- personal digital assistants (PDAs), .NET framework on, 3–4**
- point, in GDI+ classes, 1113**
- polymorphisms, 197–199**
  - interface, 198–199
  - Windows Forms Designer, 205
- portal web services, 713**
- precedence, of operators, 52–53**
  - Boolean, 68–69
- preference settings**
  - VCE, 17
  - VS 2008, 14–16
- PreviousPage, 640–641**
- PrintDialog, 570–574**
  - as addition, 571
  - options, 572
  - page ranges, 574
  - print selection, 572–574
    - as addition, 572–574
  - properties, 574
- PrintDocument, 560–565**
- printers, PageSetupDialog, 569**
- printing, 559–577**
  - architecture, 560–561
    - functionality of, 560
  - dialogs, 568–575
  - multiple pages, 565–567
  - .NET Framework, 559–577
  - PageSetupDialog, 568–570
    - as addition, 569–570
    - margins, 568
    - orientation, 569
    - paper, 568
    - printer properties, 569
  - PrintDialog, 570–574
    - as addition, 571
    - options, 572
    - page ranges, 574
    - print selection, 572–574
      - properties, 574
  - PrintDocument, 560–565
  - PrintPage event, 562–563, 565–567
    - modification, 566–567
  - PrintPreviewDialog, 575–576
    - PrintPreviewControl, 576
    - sequence, 561–562
- PrintPage event, 562–563, 565–567**
  - modifications, 566–567
  - multiple pages, 565–567
- PrintPreviewControl, 576**

## P

- padding control, 1175**
- PageSetupDialog, 568–570**
  - as addition, 569–570
  - margins, 568
  - orientation, 569
  - paper, 568
  - printer properties, 569
- Paint, 438**
- parameter arrays, 133–135**
  - application, 134
- parameter functions, for C#, 130–138**
  - arrays, 133–135
    - application, 134
  - double, 130, 135
  - global data v. return values, 143–144
  - matching, 132–133
  - out, 137–138
  - reference, 135–136

## **PrintPreviewDialog, 575–576**

PrintPreviewControl, 576

**procedural programming. See functional programming**

**profiles, in web programming, 683–688**

anonymous users, 687–688

events, 687

with components, 686

custom data types, 686

binary serialization, 686

string serialization, 686

XML serialization, 686

groups, 685–686

information creation, 684–685

properties, 684–685

**programming. See ASP.NET; C# language; functional programming; object-oriented programming techniques; web programming; Windows Forms Designer; Windows Presentation Foundation; Windows Task Manager, infinite looping**

**projection, in LINQ, 872–875**

method syntax, 875

object creation, 872–874

**properties, C# language, 244–247**

accessors, 244

addition of, 251–252

anchoring controls, 437

automatic, 253

button controls, 440

CheckBox control, 456

class members, 244–247

ColorDialog, 580

definitions, 244–247

docking control, 437

FolderBrowserDialog, 581

FontDialog, 578

indexers, 285–289

initializers, 400–401

label controls, 444

LabelTextbox, 528–529, 532

LinkLabel controls, 444

ListBox controls, 470–471

List<T>, 342

ListView control, 475–477

object initializers, 400–401

OOP techniques, 188–189

PrintDialog, 574

RadioButton control, 455

RichTextBox control, 462–463

SaveFileDialog, 555

TabControl control, 487–488

TextBox controls, 445

ToolStrip, 502

Windows Forms Designer controls, 434–435

**properties, OOP techniques, 188–189**

**properties windows, 17, 23**

VS 2008, 17

**property element syntax, 1162**

content and, 1163

**provider objects, 937–938**

command, 938

CommandBuilder, 938

connection, 937

DataAdapter, 938

DataReader, 938

**public instance methods, 538**

**publish configuration options, 589–590**

**publish wizard, 590–597**

application, 594–597

updates, 595–597

**publishing websites, 765–767**

## Q

**queries. See also Language Integrated Query; Structure Query Language server database**

group, 881–883, 914–917

LINQ to XML, 1009–1015

attributes, 1013–1015

descendants, 1011–1013

elements, 1010–1011

ordering, 914–917

select distinct, 875–876

XPath, 847

## R

**RAD. See Rapid Application Development**

**RadioButton control, 454–455**

events, 455

GroupBox control, 457–458

properties, 455

**Rapid Application Development (RAD), 433**

**ReadMe dialog, 618**

**ReadOnly, 445**

**rectangle, in GDI+ classes, 1114**

**RedoActionName, 462**

**refactoring, 252–253**

**reference parameters, 135–136**

nonconstant variables, 136

**reference types, 202**

generic, 202

nullable, 202

objects, 237

**ReferenceEquals(object,object), 216**

**reflection, 353, 1023–1026**

**regions, in GDI+ classes, 1116–1118**

**relational operators. See comparison operators**

**relative path names, 784**

**Remote Method Invocation (RMI), 711**

**remote procedure calls (RPC), 710**

**return statements, 76**

C# functions, 128–130

limitations, 129

**return values**

event handlers, 388

global variables v., 143–144

**reusable code, 125**

**ReverseString(), 415–417**

**RichTextBox control, 461–469**

events, 463–469

properties, 462–463

**RMI. See Remote Method Invocation**

**root elements, 822**

**routed events, 1168–1173**

**rows, ADO.NET, 956–964**

addition of, 956–959

deletion, 963–965

navigation of, 966–973

downstream, 973  
upstream, 973  
searching for, 959–963

**RPC. See remote procedure calls****Rtf, 462****runtime**

ASP.NET, 628  
JavaScript, 701  
WF, 1266–1273  
with persistent service, 1267–1273

**S****Sandcastle, 1072****sandcastledocs.com, 1072****SaveFileDialog, 553–559**

as addition, 555–558  
extensions, 554  
overwriting files, 554  
properties, 555  
title, 554  
forms, 558–559  
validation, 554

**schemas, XML, 824–828**

XRD, 824  
XSD, 825–828

**ScriptManager, 743****ScrollBars, 445****SDI. See Single Document Interface****sealed classes, 197, 210****SEH. See structured exception handling****select clause, 854****select distinct query, 875–876****SelectedText, 445, 462****SelectionAlignment, 462****SelectionBullet, 462****SelectionColor, 462****SelectionFont, 462****SelectionLength, 445, 462****SelectionProtected, 463****self-hosted services, 1238–1245****semantic errors, 155****SequenceActivity, 1257–1260**

cleanup code, 1259–1260  
fault handling, 1257–1259

**serialized objects, 805–810****server-side state management, 648–651**

application, 650  
caches, 651  
session, 649–650

**service calls, asynchronous, 728–731****service contracts, WCF, 1216, 1220, 1231–1238**

data, 1220, 1231  
fault, 1220, 1233  
message, 1220, 1233  
operation, 1220, 1232–1233

**service-oriented architecture (SOA), 1215****session state, ASP.NET, 649–650****set operators, 887–891****setup editors, 607****shallow copying, 238****shallow copying, deep v., 238****shape drawing, in GDI+ class, 1122–1124****short types, 95, 102****ShowSelectionMargin, 463****Simple Mail Transfer Protocol (SMTP), 1078****SimpleEditor, 599–622**

file systems editor, 607–610  
file addition, 609–610  
folder additions, 608  
properties, 608–609  
file types editor, 611–612  
action creation, 611  
extension, 612  
properties, 611  
installation, 617–622  
completion, 622  
confirmation, 621  
disk cost, 620  
license agreement, 619  
optional files, 619  
progress control, 621  
ReadMe dialog, 618  
select folder, 620  
uninstall, 622  
welcome dialog, 618  
launch condition editor, 612–613  
project creation, 603  
properties, 604–607  
packaging, 604  
prerequisites, 605  
project configuration, 606  
setup editors, 607  
user interface editor, 613–617  
dialogs, 614–617

**Single Document Interface (SDI), 493, 512–514****site navigation, 679–680**

addition of, 679–680  
controls, 679

**size, in GDI+ classes, 1114****Skip(), 883–885****SMTP. See Simple Mail Transfer Protocol****snap-in delegates, 153****snap-lines, 436–437****snapping control, 435–437**

snap-lines, 436–437

**SOA. See service-oriented architecture****SOAP, 711, 715–716**

firewalls, 716  
Windows clients, 724–728  
XML, 711

**SoapHttpClientProtocol, 719****Solution Explorer, 22**

AssemblyInfo.cs file, 22  
console applications in, 22  
overflow, 98  
VCE, 22  
VS 2008, 16–17

**spaghetti code, 69****SQL server database. See Structure Query Language server database****SQL Server .NET data provider, 939****stack order, 1174–1175****StackPanel control, 1181–1183****state management, ASP.NET, 646–651**

client-side, 646–648  
cookies, 647–648  
ViewState, 647  
server-side, 648–651

## **state management, ASP.NET (continued)**

- application state, 650
- caches, 651
- session state, 649–650

## **statement bodies, 424–425**

### **statements**

- break, 75–76
- infinite looping, 90
- C# language, 32
- default, 76
- if, 71–75
  - application, 72–74
  - conditions, 74–75
  - else blocks, 74
  - else statements w., 71
- JavaScript, 702–703
- namespace, 35
- switch, 75–78
  - application, 77–78
  - constant variables, 77
  - execution, 78
  - for loops, 88
  - structure, 75–76

## **static members, 191–192**

- UML syntax, 191–192

## **static modifiers, 148**

## **StatusStrip, 509–512**

- properties, 510–511

## **StatusStripStatusLabel, 509**

- properties, 509

## **storage, underlying types, 104**

## **stored procedures, 983–985**

## **StreamReader object, 794–801**

- data reading, 796
- delimited files, 797–801
- CSV, 797–801

## **streams, 778–805. See also FileStream object;**

### **StreamReader object**

- input/output classes, 778–805
  - compressed data, 802–805
  - directory, 780–781
  - DirectoryInfo, 783–784
  - file, 780–781
  - FileInfo, 781–783
  - FileStream object, 784–791
  - path names, 784
  - StreamReader object, 794–801
  - StreamWriter object, 791–794
- serial device, 778

## **StreamWriter object, 791–794**

## **string literal values, 44–45**

- Unicode Value, 44
- as verbatim, 44

## **string manipulation, 118–123**

- application, 120–122
- auto-completion, 120–123
  - IDE, 123
- commands, 119
- execution, 122–123
- VCE, 122–123
  - execution, 122–123
- VS 2008, 120–123
  - application, 120–122
  - execution, 122–123

## **string serialization, 686**

## **structs, 93, 108–110, 235–237**

- application, 109–110
- classes v., 236–237
- definition, 108
- execution, 110
- functions, 147–148
  - static modifiers, 148
- generic classes, 364
- objects, 202
- orientation enumerations, 109–110
- value types, 237

## **Structure Query Language (SQL) server database, 660–669**

### **ADO.NET**

- stored procedures, 983–985
- where clause, 978

### **ADO.NET support, 977–985**

- data adapter commands, 977–981
- data controls, 662
  - GridView, 663–669
- database creation, 660–661
- installation, 896–898
  - Express 2005, 897
- LINQ to SQL, 850, 895–929
- Northwind Sample database, 898–907

## **structured exception handling (SEH), 177–182**

- notes, 183–184
- throw exceptions, 180–182

## **style controls, 1184–1185**

## **suffixes, 43**

## **SuspendActivity, 1253–1254**

## **switch statements, 75–78**

- application, 77–78
- break statements, 75–76
- constant variables, 77
- default statements, 76
- execution, 78
- for loops, 88
- structure, 75–76

## **syntax**

- C# language, 32–34
  - case-sensitivity, 34
  - comments, 33
  - console application structure, 34–35
  - indentations, 32
  - statements, 32
  - white space characters, 32
- LINQ, 855–856
- lollipop, 193
- XAML, 1161–1163
  - attribute, 1161–1162
  - content, 1162–1163
  - markup extensions, 1163
  - object element, 1161
  - property element, 1162

## **System.AttributeUsageAttribute, 1041–1042**

- target values, 1041–1042

## **System.Collections, 200–201, 277–303.**

### **See also collections**

- arrays in, 277–278
- ICollection, 278, 285, 292
- IDictionary, 278, 292–294
- IEnumerable, 278, 283, 285, 292
- IList, 278, 285, 287

## **System.Collections.Generic namespace, 333–351**

- Dictionary<K,V>, 350–351

- List<T>, 341–344
  - application, 343–344
  - methods, 342
  - properties, 342
  - searching/sorting, 345–347
- System.Data namespace, 939–940**
  - native providers, 940
  - ODBC .NET data provider, 940
  - OLE DB .NET data provider, 939
  - Oracle .NET Data provider, 939
  - SQL Server .NET data provider, 939
- System.Diagnostics.ConditionalAttribute, 1027–1029**
- System.Drawing.Drawing2D, 1139–1140**
- System.Drawing.Imaging, 1140**
- System.Object, 211, 215–218**
  - constructor execution sequence, 219
  - methods, 216–217
    - Equals(object), 216
    - Equals(object,object), 216
    - GetHashCode(), 217
    - GetType(), 217
    - MemberwiseClone(), 217
    - Object(), 216
    - Object(), 216
    - ReferenceEquals(object,object), 216
    - ToString(), 217
- System.ObsoleteAttribute, 1029–1031**
- System.Reflection.AssemblyDelaySignAttribute, 1034–1038**
  - delay signing, 1035–1037
  - GAC registering, 1037–1038
  - strong name completion, 1038
- System.SerializableAttribute, 1031–1034**

## T

- TabControl control, 486–491**
  - application, 488–489
  - event handlers, 490–491
  - properties, 487–488
- TabIndex, 435**
- TabStop, 435**
- Tag, 435, 449–450**
- Take(), 883–885**
- TCP. See Transmission Control Protocol**
- TcpClient, 1096–1104**
  - client creation, 1099–1104
  - controls, 1100
- TcpListener, 1096–1104**
  - server creation, 1096–1099
- templates, 1185–1189**
- ternary operators, 46, 70–71**
- text drawing, GDI+ classes, 1124–1128**
  - font class, 1124–1128
- text editors, MDI, 517–519**
  - properties, 519
- TextBox control, 444–454**
  - application, 447–448
  - events, 446–454
  - handlers, 448–454
  - properties, 445
- texture brush class, in GDI+ class, 1131–1133**
- this keyword, 256–257**

- 3.0 enhancements, C# language, 399–429. See also initializers; Language Integrated Query**
  - anonymous types, 408–411
    - application, 409–410
    - execution, 410–411
  - automatic properties, 253
  - extension methods, 412–417
    - application, 413–415
    - definition parameters, 412–413
    - execution, 415–417
    - ReverseString(), 415–417
    - ToStringReversed(), 415–416
    - ToTitleCase(), 415, 417
  - initializers, 400–406
    - collection, 402–406
    - object, 400–402
  - lambda expressions, 418–429
    - anonymous methods, 418–423
    - collections, 426–429
    - delegates, 425–426
    - expression trees, 425–426
    - LINQ, 418–429
    - MSIL code, 419
    - parameters, 423
    - simple, 420–423
    - statement bodies, 424–425
  - LINQ, 9, 417–429, 849–892
    - aggregate operators, 865–868
    - Boolean values, 877–878
    - complex objects, 868–872
    - First(), 886–887
    - first query, 850–854
    - FirstOrDefault(), 886–887
    - group queries, 881–883
    - joins, 891–892
    - lambda expressions, 418–429, 856–858
    - large data sets, 862–864
    - methods, 855–856
    - orderby clause, 860
    - ordering in, 860–862
    - projection, 872–875
    - query results, 858–859
    - resources, 892
    - select distinct query, 875–876
    - set operators, 887–891
    - Skip(), 883–885
    - Take(), 883–885
    - varieties, 850
    - partial methods, 264–265
    - type inference, 406–407
  - throw exceptions, 180–182**
  - timer controls, 746–747**
  - title setting**
    - OpenFileDialog, 541
    - SaveFileDialog, 554, 558–559
    - for forms, 558–559
  - toolbars, 501–512**
    - event handlers, 505–508
    - StatusStrip, 509–512
      - properties, 510–511
    - ToolStrip, 502–509
      - extension, 503–505
      - items, 503
      - properties, 502
    - VS 2008, 16, 501–512

**ToolsStrip, 502–509**  
extension, 503–505  
items, 503  
properties, 502

**ToolStripMenuItem, 496–499**

**ToString(), 217**

**ToStringReversed(), 415–416**

**ToTitleCase(), 415, 417**

**tracepoints, 163–165**  
diagnostics output v., 165–166

**Transmission Control Protocol (TCP), 1078, 1081–1083**  
addresses, 1218  
application protocols, 1081–1083  
WCF, 1217

**triggers, 1189–1191**

**try, as keyword, 177–178**

**Two's complement, 64**

**type comparisons, 303–308**  
boxing, 304–305  
is operator, 306–308  
application, 306–308  
execution, 308  
unboxing, 304–305

**type conversions, 51, 93–103. See also explicit conversions**  
application, 101–102  
execution, 102–103  
float values, 102  
explicit, 95–103  
code modification, 96  
covert commands, 100–103  
definition, 96  
overflow checking, 97–99  
short values into bytes, 96  
implicit, 47, 94–95  
char types, 93–94  
code, 94  
short types, 95, 102  
storage values, 94

**typeof operators, 107**

**types, in C# language, 36–41. See also type conversions**  
anonymous, 408–411  
application, 409–410  
execution, 410–411  
application of, 39  
assignment operator, 40  
bits, 36  
Boolean, 36, 38  
execution of, 39–41  
floating-point values, 37  
integer, 37  
OOP techniques, 202  
reference, 202  
value, 202  
simple, 36–41  
3.0 enhancements, 406–407

## U

**UDP. See User Datagram Protocol**

**UML. See Universal Modeling Language**

**unary operators, 46–48**  
overloading, 312

**unbounded types, 355**

**unboxing, 304–305**

**underlying types, 104**

**UndoActionName, 463**

**Unicode Value, 44**

**Uniform Resource Identifier (URI), 823**  
networking, 1080–1081

**Universal Modeling Language (UML), 187**  
containment, 200  
inheritance, 195  
lollipop syntax, 193  
static members, 191–192

**unsigned integer types, 37**

**update panels, 741–746**  
properties, 746  
ScriptManager, 743  
triggers, 744–746

**update progress controls, 747–749**

**upstream navigation, 973**

**URI. See Uniform Resource Identifier**

**user controls, 524–535**  
debugging, 530–531  
LabelTextbox, 526–530  
event handler addition, 529–530, 532–534  
extension, 531–535  
property addition, 528–529, 532  
web programming, 680–683  
creation, 681–682  
use of, 683  
WPF programming, 1200–1201

**User Datagram Protocol (UDP), 1078, 1081–1083**  
application protocols, 1081–1083

**user interface editor, 613–617**  
dialogs, 614–617  
addition of, 615–617  
default, 614–615

## V

**Validated/Validating, 438**

**validation, of dialogs, OpenFileDialog, 543–544**

**value comparisons, 303, 309–325**  
icomparable interface, 320–325  
icomparer interface, 320–325  
operator overloading, 309–320  
binary, 312  
CardLib, 314–320  
unary, 312

**value parameters, 135–136**

**value types, 202**  
structs, 237

**variable declaration, 45–46**

**variable scope, 138–144**  
branching, 141  
definition, 138–141  
global, 140  
local, 140  
looping, 141  
operations, 139  
out of scope, 139  
use of, 138–141

**variables, 36–56, 93–123. See also conventions, naming; mathematical operators; operators; string manipulation; type conversions**  
arrays, 93, 111–118  
application, 113

- declaration of, 111–112
  - elements, 111
  - execution, 113–114
  - foreach loops, 114–116
  - initialization, 112
  - jagged, 116–117
  - multidimensional, 115–116
  - single base types, 111
  - size matching, 112
  - assignments, 36, 45–46
  - constant, 77
  - declaration of, 45–46
  - enumerations, 93, 103–118
    - application, 105–106
    - definition, 103–105
    - execution, 106–107
    - orientation types, 103, 106–107
    - typeof operators, 107
    - underlying types, 104
  - expressions, 46–56
    - operators, 46–53
  - literal values, 43–45
    - string, 44–45
    - suffixes, 43
  - namespaces, 53–56
  - naming, 41–43
    - conventions, 42–43
  - nonconstant, 136
  - scope of, 138–144
    - branching, 141
    - definition, 138–141
    - global, 140
    - local, 140
    - looping, 141
    - operations, 139
    - out of scope, 139
    - use of, 138–141
  - string manipulation, 118–123
    - application, 120–122
    - auto-completion, 120–123
    - commands, 119
    - execution, 122–123
    - VCE, 122–123
    - VS 2008, 120–123
  - structs, 93, 108–110
    - application, 109–110
    - definition, 108
    - execution, 110
    - orientation enumerations, 109–110
  - types, 36–41
    - application of, 39
    - assignment operator, 40
    - bits, 36
    - Boolean, 36, 38
    - complex, 103–118
    - conversions, 51, 93–103
    - execution of, 39–41
    - floating-point values, 37
    - integer, 37
    - simple, 36–41
- VCE. See Visual C# 2008 Express Edition**
- ViewState, 647**
- virtual classes, 196**
- Visual Basic language, .NET Framework compatibility with, 4**
- Visual C# 2008 Express (VCE) Edition. See also debugging; errors, handling of; Language Integrated Query; LINQ to SQL**
- debugging, 156–176
    - break mode, 166–176
    - call stack window, 176
    - configurations, 156
    - directories, 156
    - nonbreak mode, 157–166
    - stepping through code, 174–175
    - symbolic information, 156
    - variable content monitoring, 171–174
  - error handling, 155, 176–184
    - configuring exceptions, 182
    - fatal, 155
    - listing exceptions, 182
    - logic/semantic, 155
    - namespaces, 177
    - SEH, 177–182
  - error handling of, definition, 176
  - Intellisense feature, 149
  - LINQ, 9, 417–429, 849–892
    - aggregate operators, 865–868
    - Boolean values, 877–878
    - complex objects, 868–872
    - First(), 886–887
    - first query, 850–854
    - FirstOrDefault(), 886–887
    - group queries, 881–883
    - joins, 891–892
    - lambda expressions, 418–429, 856–858
    - large data sets, 862–864
    - methods, 855–856
    - orderby clause, 860
    - ordering in, 860–862
    - projection, 872–875
    - query results, 858–859
    - resources, 892
    - select distinct query, 875–876
    - set operators, 887–891
    - Skip(), 883–885
    - Take(), 883–885
    - varieties, 850
  - OOP techniques, 223–229
    - class additions, 226–227
    - Class View window, 223–225
    - diagrams, 227–229
    - Object Browser, 225–226
  - products, 10
  - string manipulation, 120–123
    - application, 120–122
    - execution, 122–123
  - Windows Forms Designer, 13, 25–28, 433–491
    - code outlining, 27–28
    - controls, 434–491
    - creation, 25–26
    - execution, 26–28
    - Form1.cs file, 27–28
    - Form1.Designer code, 27–28
    - OOP techniques, 203–206
  - writing programs, 13–29
    - code outlining, 27–28
    - console applications, 18–21
    - error list window, 24
    - preference settings, 17

## Visual C# 2008 Express (VCE) Edition (continued)

- solution Explorer, 22
- XAML, 1145–1147

## Visual Studio (VS) 2008. *See also controls, Visual Studio 2008; debugging; deployment, of web applications; errors, handling of; Language Integrated Query; LINQ to SQL; Multiple Document Interface; SimpleEditor; structured exception handling; user controls*

- breakpoints, 168
- controls, 524–535
  - custom, 524
  - user, 524–535
- debugging, 156–176
  - break mode, 166–176
  - call stack window, 176
  - configurations, 156
  - directories, 156
  - nonbreak mode, 157–166
  - stepping through code, 174–175
  - symbolic information, 156
  - variable content monitoring, 171–174
- deployment project types, 597–598
- error handling, 155, 176–184
  - configuring exceptions, 182
  - definition, 176
  - fatal, 155
  - listing exceptions, 182
  - logic/semantic, 155
  - namespaces, 177
  - SEH, 177–182
- features, 9–10
- IDE, 9, 14–17
  - error list, 17
  - layout, 16
  - menus, 16
  - preference settings, 14–16
  - properties window, 17, 23
  - solution Explorer, 16–17
  - toolbars, 16
- Intellisense feature, 149
- JavaScript, 701
- LINQ, 9, 417–429, 849–892
  - aggregate operators, 865–868
  - Boolean values, 877–878
  - complex objects, 868–872
  - First(), 886–887
  - first query, 850–854
  - FirstOrDefault(), 886–887
  - group queries, 881–883
  - joins, 891–892
  - lambda expressions, 418–429, 856–858
  - large data sets, 862–864
  - methods, 855–856
  - orderby clause, 860
  - ordering in, 860–862
  - projection, 872–875
  - query results, 858–859
  - resources, 892
  - select distinct query, 875–876
  - set operators, 887–891
  - Skip(), 883–885
  - Take(), 883–885
  - varieties, 850
- MDI, 493, 512–524
  - applications, 514–516

- text editors, 517–519
- tracking, 522–524
- menus, 16, 494–501
  - functionality additions, 499–501
  - manual creation, 495–498
  - menustrip control, 495–499
  - merging, 519–522
  - text editors, 519–522
- OOP techniques, 223–229
  - class additions, 226–227
  - Class View window, 223–225
  - diagrams, 227–229
  - Object Browser, 225–226
- SDI, 493, 512–514
- SimpleEditor, 599–622
  - file systems editor, 607–610
  - file types editor, 611–612
  - installation, 617–622
  - launch condition editor, 612–613
  - project creation, 603
  - properties, 604–607
  - user interface editor, 613–617
- string manipulation, 120–123
  - application, 120–122
  - execution, 122–123
- toolbars, 16, 501–512
  - event handlers, 505–508
  - StatusStrip, 509–512
  - ToolStrip, 502–509
- tracepoints, 163–165
- web application deployment, 759–772
  - copying websites, 763–765
  - IIS, 628
  - publishing websites, 765–767
  - Windows Installer, 767–772
- Windows Forms Designer, 13, 25–28, 433–491
  - code outlining, 27–28
  - controls, 434–491
  - execution, 26–28
  - Form1.cs file, 27–28
  - Form1.Designer code, 27–28
  - OOP techniques, 203–206
- Windows Installer, 598–622
  - advantages, 601
  - architecture, 598–601
  - component, 599
  - feature, 599
  - package, 599
- writing programs, 13–29
  - code outlining, 27–28
  - console applications, 18–21
  - IDE, 14–17
  - XAML, 1145–1147
  - XML documents, 826–828

## VS 2008. *See Visual Studio 2008*

## W

- [w3.org/TR/SOAP](http://w3.org/TR/SOAP), 711
- WAS.** *See Windows Activation Service hosting*
- WCF.** *See Windows Communication Foundation*
- weather.com**, 711
- weather.msn.com**, 711
- web parts, web programming**, 688–701

- application from, 690–692
- catalog zone, 695–697
  - addition of, 697
- connection zone, 697–701
  - creation of, 698–701
  - properties, 699
- editor zone, 692–695
  - addition of, 694–695
  - appearances, 693
  - behavior, 693
  - layout, 692–693
- WebPartManager, 689
- WebPartZone, 689–690
- web programming, 671–707. See also ASP.NET; deployment, of web applications; profiles, in web programming; web parts, web programming**
- application deployment, 759–772
- copying websites, 763–765
  - IIS, 628
  - publishing websites, 765–767
  - Windows Installer, 767–772
- JavaScript, 32, 701–707
  - controls, 706–707
  - CustomValidator, 703–704
  - function definition, 702
  - objects, 703–705
  - runtime, 701
  - script element, 701
  - statements, 702–703
  - variable declaration, 702
  - window opening, 705–706
- master pages, 671–678
  - application, 675–676
  - creation, 673–676
  - use of, 677–678
- profiles, 683–688
  - anonymous users, 687–688
  - with components, 686
  - custom data types, 686
  - groups, 685–686
  - information creation, 684–685
  - properties, 684–685
- site navigation, 679–680
  - addition of, 679–680
  - controls, 679
- user controls, 680–683
  - creation, 681–682
  - use of, 683
- web parts, 688–701
  - application from, 690–692
  - catalog zone, 695–697
  - connection zone, 697–701
  - editor zone, 692–695
  - WebPartManager, 689
  - WebPartZone, 689–690
- web server hosting, 1221**
- Web Service Enhancements (WSE), 1215**
- web services, 709–736. See also deployment, of web applications**
- AJAX, 749–754
  - WCF, 749–754
- application deployment, 759–772
  - copying websites, 763–765
  - IIS, 628
  - publishing websites, 765–767
  - Windows Installer, 767–772
- application-specific, 713
- application-to-application, 711–712
- architecture, 713–717
  - methods, 714–716
  - WS-I Basic Profile specification, 717
- ASP.NET, 720–722
  - client implementation, 731–732
  - method addition, 721–722
- asynchronous service calls, 728–731
- book distributor applications, 712
- building block, 713
- client application types, 713
- CORBA, 710
- data passage, 732–736
- DCOM, 711
- firewalls, 716
- hotel travel agency applications, 712
- .NET Framework, 717–720
  - alternative client protocols, 719–720
  - clients, 719
  - properties, 717
  - SoapHttpClientProtocol, 719
  - WebMethod attribute, 718
  - WebService attribute, 718
  - WebServiceBinding attribute, 719
- portal, 713
- RMI, 711
- RPC, 710
- SOAP, 711, 715–716
- testing applications, 722–723
- user-to-application, 711–712
- Windows clients, 724–728
- WPF, 1164
- WebClient, 1084–1086**
  - properties, 1084–1086
- WebMethod attribute, 718**
- WebPartManager, 689**
- WebPartZone, 689–690**
- WebRequest, 1084–1095**
  - controls, 1088
- WebResponse, 1084–1095**
  - controls, 1088
- WebService attribute, 718**
- WebServiceBinding attribute, 719**
- websites**
  - ajax.asp.net, 740
  - application deployment, 763–765
    - copying, 763–765
    - publishing, 765–767
  - ndoc.sourceforge.net, 1071
  - sandcastledocs.com, 1072
  - w3.org/TR/SOAP, 711
  - weather.com, 711
  - weather.msn.com, 711
- welcome dialog, 618**
- WF. See Windows Workflow Foundation**
- where clause, 853**
  - ADO.NET, 978
- while loops, 81–83**
  - application, 82
  - Boolean test, 81
  - do v., 83
  - execution, 83

## while loops (continued)

---

### while loops (continued)

- false evaluation, 81
- user validation, 83

### WhileActivity, 1254–1256

### White, Eric, 892

### white space characters, 32

### Windows Activation Service (WAS) hosting, 1221

### Windows applications deployment. *See* deployment, of Windows applications

### Windows Communication Foundation (WCF), 749–754, 1215–1245

- addresses, 1218–1220
- behaviors, 1221
- bindings, 1216, 1218–1220
- definition, 1216–1217
- endpoints, 1218–1220
- hosting, 1221
- message patterns, 1220–1221
- programming, 1221–1230
- protocols, 1217–1218
  - HTTP, 1217
  - MSMQ, 1217
  - named pipe, 1217
  - TCP, 1217
- self-hosted services, 1238–1245
- service contracts, 1216, 1220, 1231–1238
  - data, 1220, 1231
  - fault, 1220, 1233
  - message, 1220, 1233
  - operation, 1220, 1232–1233
- SOA, 1215

### Windows Forms Designer, 13, 24–28, 433–491.

*See also* controls, Windows Forms Designer

- code outlining, 27–28
- controls, 434–491
  - properties, 434–435
- creation, 25–26
- execution, 26–28
- Form1.cs file, 27–28
- Form1.Designer code, 27–28
- OOP techniques, 203–206
  - polymorphisms, 205
- VCE, 13, 25–28, 433–491
- VS 2008, 13, 25–28, 433–491

### Windows Installer, 598–622, 767–772. *See*

*also* deployment, of Windows applications;

#### SimpleEditor

- advantages, 601
- architecture, 598–601
  - terms, 599–601
- component, 599
- feature, 599
- installation, 769–772
- package, 599
- properties, 769
- setup program, 767–769
- SimpleEditor, 599–622
  - file systems editor, 607–610
  - file types editor, 611–612
  - installation, 617–622
  - launch condition editor, 612–613
  - project creation, 603
  - properties, 604–607
  - user interface editor, 613–617

### Windows operating system. *See also* C# language; .NET Framework; Windows Communication Foundation; Windows Forms Designer; Windows Installer; Windows Presentation Foundation; Windows Workflow Foundation

- C# language, 8
  - ADO.NET, 9
  - applications, 8–9
  - as block-structured, 32
  - C++ v., 8
  - class definitions, 195–197, 209–238
  - class members, 241–276
  - debugging, 155–176
  - error handling, 155, 176–184
  - flow control, 59–90
  - functions, 125–154
  - generics, 331
  - interfaces, 193–194, 198–199, 212–215, 233–235
  - LINQ capabilities, 9
  - methods, 242–244, 250–251, 254–256, 264–265, 388, 412–417, 471, 477
  - OOP techniques, 4, 185–206
  - syntax, 32–34
  - System.Object, 211, 215–218
  - as type-safe, 8
  - variables, 36–56
  - writing programs, 13–29
- .NET Framework, 3–4
  - ASP.NET, 627–669
  - C# language compatibility, 4
  - C++ language compatibility, 4
  - class definitions, 193–197, 209–238, 257–261
  - class members, 241–276
  - COBOL language compatibility with, 4
  - destructors, 191
  - dialogs, 537–538
  - generics, 332–351
  - IComparer interface, 321
  - JavaScript language compatibility, 4
  - library of, 4
  - Linux operating system, 3
  - Mono operating system, 3
  - no-touch deployment, 584
  - OOP techniques, 4, 185–206, 223–229
  - PascalCase, 243
  - PDAs, 3–4
  - printing, 559–577
  - System.Object, 211, 215–218
  - Two's complement, 64
  - web services, 717–720
- Windows Forms Designer, 13, 24–28, 433–491
  - code outlining, 27–28
  - controls, 434–491
  - creation, 25–26
  - execution, 26–28
  - Form1.cs file, 27–28
  - Form1.Designer code, 27–28
  - OOP techniques, 203–206
- Windows Presentation Foundation (WPF), 1143–1212. *See also* Extensible Application Markup Language
  - application anatomy, 1148–1160
  - elements, 1155–1158
  - C# developers, 1148
  - control basics, 1165–1173
    - attached event, 1173

- attached properties, 1168
  - dependency properties, 1166–1167
  - routed events, 1168–1173
  - control layout, 1174–1184
    - alignment, 1175
    - border, 1175
    - canvas, 1175–1176
    - dimensions, 1175
    - DockPanel, 1177–1179
    - grid control, 1179–1181
    - margins, 1175
    - padding, 1175
    - stack order, 1174–1175
    - StackPanel, 1181–1183
    - WrapPanel, 1183–1184
  - control styling, 1184–1189
    - styles, 1184–1185
    - templates, 1185–1189
  - definition, 1144–1148
  - fundamentals, 1160–1200
    - animations, 1191–1192
    - application objects, 1164–1165
    - desktop/web applications, 1164
    - dynamic resources, 1195
    - referencing style resources, 1195–1196
    - static resources, 1194
    - syntax, 1161–1163
  - programming with, 1200–1212
    - dependency properties, 1201–1212
    - user controls, 1200–1201
  - triggers, 1189–1191
  - XAML, 1145–1147
  - Windows programming. See C# language; object-oriented programming techniques; Visual C# 2008 Express Edition; Visual Studio 2008; Windows Communication Foundation; Windows Forms Designer; Windows Installer; Windows Presentation Foundation; Windows Workflow Foundation**
  - Windows Task Manager, infinite looping, 90**
  - Windows Workflow Foundation (WF), 1247–1278.**
    - See also custom activities, in WF;**
    - SequenceActivity**
  - activities, 1251–1266
    - custom, 1260–1266
    - DelayActivity, 1251–1252
    - SequenceActivity, 1257–1260
    - SuspendActivity, 1253–1254
    - WhileActivity, 1254–1256
  - data binding, 1273–1278
    - bound property definitions, 1274–1278
    - workflow runtime, 1266–1273
      - with persistent service, 1267–1273
  - WordWrap, 445**
  - WrapPanel control, 1183–1184**
  - writing programs, in C# language, console applications, 13, 18–21, 34–35**
    - structure, 34–35
  - WSE. See Web Service Enhancements**
  - WS-I Basic Profile specification, 717**
- ## X
- XAML. See Extensible Application Markup Language**
  - XDR schemas. See XML-Data Reduced schemas**
  - XML. See Extensible Markup Language**
  - XML document object model (XML-DOM), 828–838**
    - classes, 828–831
      - XmlDocument, 829
      - XmlElement, 829–831
    - nodes, 831–839
      - creation of, 834–837
      - deletion of, 838–839
      - looping, 831–833
      - properties, 834–835
      - value changes, 833–836
  - XML documentation. See documentation, XML**
  - XML documents. See documents, XML**
  - XML Schema Definition (XSD), 825–828**
  - XML schemas. See schemas, XML**
  - XML-Data Reduced (XDR) schemas, 824**
  - XmlDocument class, 829**
  - XML-DOM. See XML document object model**
  - XmlElement class, 829–831**
    - properties, 830
  - XPath nodes, 839–847**
    - common operations, 840
    - queries, 847
  - XSD. See XML Schema Definition**
  - XSLT, XML documentation with, 1070–1071**

