

Index

Symbols

/App Code folder, code-beside/
 partial classes page
 compilation, 101-103
 /bin folder, 7
 @Control directive, 127
 @Page directive, 127
 @WebHandler directive, 127

A
 abstract classes, defined, 32-33
 access modifiers, defined, 18
 ACT (Application Center Test),
 258-259, 265
 action attribute, 108
 Add Reference dialog, 145, 243
 Add Web Reference dialog, 173
 AddWithValue() method, 51, 71
 agile programming, defined, 35
 AmazonSearchService, 171
 anonymousIdentification tag, 208
 /App Code folder, code-beside/
 partial classes page
 compilation, 101-103
 Append() method, 255
 application architecture
 defined, 45
 layers
 advantages of, 45-47
 n-tier architecture, 47-48
 POP Forums architecture
 case study, 48-59
 requirements assessment, 60
 service-oriented architecture
 (SOA), 59
 Application Center Test. *See* ACT
 application events, 111-113
 application performance. *See*
 performance

ApplicationName property
 Membership provider, 200
 ProfileProvider, 209
 AppSettings prefix (default
 expression), 235
 architecture. *See* application
 architecture
 ArrayList class, 83-87
 aspnet_compiler.exe command
 line tool, 103
 aspnet_Profile table, 213
 aspnet_regsql.exe utility, 205
 aspnet_Users table, 213
 assemblies, defined, 7
 Attach to Process dialog, 277
 attributes
 action, 108
 BufferResponse, 169
 CacheDuration, 169
 DeleteCommand, 232
 Description, 169
 EnableSession, 169
 enableViewState, 254
 MessageName, 169
 OnClick, 114
 onsubmit, 108
 pollTime, 252
 SkinID, 224
 Test, 270
 TestFixture, 270
 Theme, 221
 TransactionOption, 169
 UpdateCommand, 232
 WebMethod, 166
 AuthenticateRequest event, 135
 AutoPostBack property, 144

B

bandwidth leeching
 (HttpHandlers), 122-128
 /bin folder, 7
 break points, 246
 BufferResponse attribute, 169
 building server controls, 145-151
 built-in membership provider,
 186-189
 built-in server controls, 139-145
 business rules layer
 connecting to user interface
 layer, 57-58
 creating, 55-57
 ButtonClickHandler() event
 handler, 159

C

CacheDependency object, 251
 CacheDuration attribute, 169
 caching, 265
 classes, 53-55
 data caching, 250-253, 265
 custom data classes, 77-78,
 81-82
 fragment caching, 250
 Calendar control, 111
 car analogy, 7-9
 case study, POP Forums
 architecture, 48-59
 ChangePassword() method, 185
 ChangePasswordQuestionAnd
 Answer() method, 185
 class members
 defined, 17
 naming conventions, 21
 scope, defined, 26-27
 classes. *See also* custom data
 classes; objects
 abstract classes, defined, 32-33
 access modifiers, defined, 18

302 Index

- caching, 53-55
 - car analogy, 7-9
 - constructors, defined, 19-20
 - Control, 111
 - declaring, 18-19
 - defined, 5-7
 - designing. *See* designing classes
 - enumerations, defined, 33-34
 - EsteemBooster, 171
 - ExpressionBuilder, 235
 - FileStream, 285
 - HttpHandlers, 117
 - IHandler interface, 119-120
 - machine.config file, 117-119
 - protecting images from
 - bandwidth leeching, 122-128
 - HttpModules, 117, 128-130
 - adding users to roles, 130, 134, 137
 - HttpRequest, 291
 - HttpResponse, 291
 - inheritance, 9-15
 - instantiating, 5-6
 - interfaces, defined, 30-31
 - Membership, static methods, 184
 - MembershipUser, 185
 - methods
 - defined, 23-26
 - overloading, 27-28
 - static methods, 28-29
 - networking, 290-292, 295-296
 - partial classes, 57
 - private variables, defined, 21-23
 - ProfileInfo, 218
 - ProfileInfoCollection, 219
 - properties, defined, 21-23
 - proxy, modification, 180
 - Roles, static methods, 185
 - server controls, 139, 163
 - building controls, 145-151
 - composite controls, 156, 159-163
 - derived controls, 139-145
 - rendering HTML, 155
 - viewstate, 155
 - SqlCacheDependency, 251
 - StreamReader, 288
 - StreamWriter, 288
 - TcpClient, 290
 - TcpListener, 290
 - UdpClient, 290
 - XmlTextWriter, 162
 - Close() method (FileStream objects), 288
 - code management, testing and deployment, 267-280
 - TDD, 268-278
 - Visual Source Safe, 279
 - xcopy deployment, 278
 - code-behind, compiled page compilation, 99
 - code-behind, non-compiled page compilation, 97-99
 - code-beside/partial classes page compilation, 100-103
 - code-behind WebService directive, 167
 - CompareValidator control (declarative programming), 228
 - compiled code versus interpreted code, 7
 - compiled WebService directive, 167
 - composite server controls, 156, 159-163
 - ConnectionString prefix (default expression), 235
 - constructors
 - custom data classes, 69-73
 - defined, 19-20
 - passing parameters to, 41
 - consumption, Web services, 173-174, 177-179
 - Control class, 111
 - @Control directive, 127
 - control events, 111-113
 - controls
 - authentication, 202-203
 - Login, 202
 - LoginName, 202
 - LoginStatus, 202
 - LoginView, 131, 203
 - PasswordRecovery, 203
 - declarative programming, 227-228
 - data controls, 229-232
 - navigation controls, 233-234
 - validation controls, 228-229
 - Web Parts, 232
 - converting code to Web services, 166-173
 - Copy Website window, 278
 - Create() method, 73-74
 - CreateChildControls() method, 156
 - custom data classes, 61-64, 69-71, 83-87
 - ArrayList class, 8-87
 - constructors, 69-73
 - Create() method, 73-74
 - data caching, 77-78, 81-82
 - Delete() method, 76-77
 - design requirements, 63-64
 - properties, 64, 68-69
 - Update() method, 75-76
 - custom providers, 189-195, 198-201
 - CustomerID property, custom data classes, 68
 - Customers table, 63
 - customization
 - profile providers (profile system), 209-213, 216-220
 - server controls, 145-151
 - CustomValidator control (declarative programming), 228

D

- data access layer, loading, 53-54
- data access methods, 254-255
- data caching, 250-253, 265
 - custom data classes, 77-78, 81-82
- data classes, 61-64, 69-71, 83-87
 - ArrayList class, 83-87
 - constructors, 69-73
 - Create() method, 73-74
 - data caching, 77-78, 81-82
 - Delete() method, 76-77
 - design requirements, 63-64
 - object-oriented programming, 61
 - properties, 64, 68-69
 - Update() method, 75-76
- data controls, declarative programming, 229-232
- data structures, designing classes as, 42
- databases
 - inserting data, 50-51
 - retrieving records, 3-4
- DataGrid control (declarative programming), 229

- DataSets, 254
 - Datye, Vandana (*Developing Microsoft ASP.NET Server Controls and Components*), 163
 - debugger, Visual Studio, 246-247
 - declarative programming,
 - data controls, 229-232
 - expressions, 235
 - navigation controls, 233-234
 - validation controls, 228-229
 - Web Parts, 232
 - declaring
 - classes, 18-19
 - private variables, 21
 - default behavior, Visual Studio, 241
 - default constructor, 70
 - delegates
 - ParameterizedThreadStart, 298
 - System.EventHandler, 113
 - ThreadStart, 298
 - TimerCallback, 300
 - Delete() method, 76-77
 - DeleteCache() method, 82
 - DeleteCommand attribute, 232
 - DeleteInactiveProfiles() method (profile providers), 219
 - DeleteProfiles() method (profile providers), 219
 - deployment, 267-280
 - TDD, 268-278
 - Visual Source Safe, 279
 - xcopy deployment, 278
 - derived server controls, 139-145
 - Description attribute, 169
 - Description property (WebService attribute), 170
 - design requirements, custom
 - data classes, 63-64
 - designing classes
 - as data structures, 42
 - focus narrowing phase, 36-37
 - methods versus properties, 37-41
 - requirements gathering phase, 35-36
 - static methods, 41-42
 - test-driven development (TDD), 43-44
 - Developing Microsoft ASP.NET Server Controls and Components*, 116, 163
 - development, code testing and
 - deployment, 267-280
 - TDD, 268-278
 - Visual Source Safe, 279
 - xcopy deployment, 278
 - development products. *See* Visual Studio
 - dialog boxes
 - Add Reference, 145, 243
 - Add Web Reference, 173
 - Attach to Process, 277
 - directives
 - OutputCache, 250
 - Register, 141
 - Dispose() method, 130
 - Disposed event, 110
 - DNS (Domain Name Service), 290
 - doPostBack client-side script, 109
- E**
- EnableSession attribute, 169
 - enableViewState attribute (Page directive), 254
 - EnableViewState property (controls), 254
 - encapsulation, defined, 5
 - Encoding property, 288
 - enumerations, defined, 33-34
 - EsteemBooster class, 171
 - event handlers, 110-111
 - ButtonClickHandler(), 159
 - EventArgs parameter, 113
 - EVENTARGUMENT field, 108
 - events, 105, 116
 - application events, 111-113
 - AuthenticateRequest, 135
 - control events, 111-113
 - history, 105, 107
 - page events, 111-113
 - postback, 113-114
 - postback process, 107-110
 - server controls, 110-111
 - wiring, 114-115
 - EVENTTARGET field, 108
 - ExecuteNonQuery() method, 76
 - execution plan (Query Analyzer), 262
 - express products, 240
- ExpressionBuilder class, 235
- expressions, declarative programming, 235
- Extensible Application Markup Language. *See* XAML
- extensions, IIS handling, 95
- extreme programming, defined, 35
- F**
- file extensions, IIS handling, 95
 - File Transfer Protocol. *See* FTP
 - FileAccess parameter, 285
 - FileMode parameter, 285
 - FileStream class, 285
 - FindInactiveProfilesByUser
 - Name() method (profile providers), 221
 - FindProfilesByUserName()
 - method (profile providers), 221
 - focus narrowing phase (designing classes), 36-37
 - form element, 108
 - form fields, hidden, 108
 - forms authentication,
 - 183-185, 203
 - built-in membership provider, 186-189
 - controls, 202-203
 - custom providers, 189-195, 198-201
 - fragment caching, 250
 - FTP (File Transfer Protocol), 290
- G**
- GAC (global assembly cache), 102
 - Get... methods, 142
 - GetAllInactiveProfiles() method (profile providers), 221
 - GetAllProfiles() method (profile providers), 220
 - GetNumberOfInactiveProfiles()
 - method (profile providers), 220
 - GetPassword() method, 185
 - GetProfile() method, 212
 - GetPropertyValues() method, 212, 216
 - GetResponse() method, 295
 - GetRoles() method, 131
 - global assembly cache. *See* GAC

304 Index

H

hardware, performance, 261
 history, events, 105-107
 HTML rendering, server
 controls, 155
 HTTP (HyperText Transfer Protocol), 290
 HTTP requests, IIS, 91, 95, 104
 file extensions, 95
 page compilation, 95-103
 pre-compiled code
 deployment, 103-104
 HttpContext class, Profile
 property, 205-208
 HttpForbiddenHandler, 119
 HttpHandlers, 30, 117
 IHttpHandler interface,
 119-120
 machine.config file, 117-119
 protecting images from band-
 width leeching, 122-128
 HttpModules, 117, 128-130
 adding users to roles, 130,
 134, 137
 HttpRequest class, 291
 HttpResponse class, 291
 HyperText Transfer Protocol. *See*
 HTTP

I

IDE (integrated development environment), 239. *See also* Visual Studio
 IETF (Internet Engineering Task Force), 291
 IHttpHandler interface, 30
 writing HttpHandlers, 119-120
 IHttpModule interface, 129
 IIS (Internet Information Services), 91, 94, 104
 file extensions, 95
 HTTP requests, 91, 95
 page compilation, 95
 code-behind, compiled, 99
 code-behind, non-compiled,
 97-99
 code-beside/partial classes,
 100-103
 inline code, 96-97
 pre-compiled code
 deployment, 103-104

IIS Manager, 91
 indexed columns, 264
 inheritance, 9-13
 from System.Object, 14-15
 Init event, 110
 Init() method, 130
 Initialize() method, 193, 209
 InitializeComponent()
 method, 115
 inline code, page compilation,
 96-97
 inline WebService directive, 167
 inserting data into databases,
 50-51
 instances, defined, 5
 instantiating
 classes, 5-6
 objects, 19-20
 integrated development
 environment. *See* IDE
 interfaces, 30-31, 52
 International Organization for
 Standardization. *See* ISO
 Internet Engineering Task
 Force. *See* IETF
 Internet Information Services.
 See IIS
 interpreted code versus compiled
 code, 7
 IPostBackDataHandler
 interface, 146
 ISO (International Organization
 for Standardization), 290
 IsReusable property
 (HttpHandlers), 121

J-K

JUnit, 268
 Kothari, Nikhil (*Developing
 Microsoft ASP.NET
 Server Controls and
 Components*), 163

L

Label controls, 11
 layers (application architecture), 47
 advantages of, 45-47
 defined, 45
 n-tier architecture, 47-48
 POP Forums architecture case
 study, 48-59
 requirements assessment, 60

leeching, protecting images
 from bandwidth leeching,
 122-128
 LinkButton, 110
 listings
 abstract base class and class
 inheriting it, 32
 adding a property to Render()
 method, 148
 anti-image leeching
 HttpHandler, 124
 assigning value to Label
 control, 11-12
 calling classes with
 properties, 39
 class declaration, 18
 class designed to pass test, 273
 class in action, 62
 code-behind, non-compiled
 page compilation, 98
 code-beside files, 100-101
 composite control, 158
 comprehensive test to supple-
 ment smaller tests, 276
 constructor, 20
 with parameters, 20, 41
 populating DropDownList
 with data, 142
 Create() method, 56-57, 73-74
 creating
 class that inherits from
 UserControl, 12
 SMTP exchange in code, 295
 Timer in HttpModule, 299
 custom control, 154
 custom server control, 147
 declaratively creating editable
 table with GridView and
 SqlDataSource, 232
 declaring private variables, 21
 default constructor, 70
 Delete() method, 76-77
 doPostBack client-side
 script, 109
 enumeration for car types, 33
 EsteemBooster class, 173
 GetPropertyValues()
 method, 216
 grouping Profile properties, 207
 "Hello world" Web service, 167
 hidden form fields, 108

- httpHandlers section of
 - machine.config, 119
- HttpModule to determine
 - user roles, 134
- httpModules section of
 - machine.config, 129
- IHttpHandler interface, 30
 - class implementing, 120
- IHttpModule, class
 - implementing, 130
- image generation HttpHandler
 - image.ashx, 127
- implementation
 - event and event handlers, 151
 - IHttpHandler interface, 31
 - Initialize() method
 - MembershipProvider, 195
 - Profile provider, 211
 - inline code, 96
 - inserting data into database,
 - 50-51
 - instantiating
 - classes, 5-6
 - classes with parameters, 41
 - objects, 19
 - objects with parameters, 20
 - IPopForumsData interface, 52
 - launching a new thread, 297
 - LinkButton, 110
 - loading data access layer, 53-54
 - LoadPostData() implemented
 - in class, 150
 - LoginView control, 203
 - manipulating
 - class without properties, 40
 - pages in code-behind
 - file, 288
 - methods
 - with no parameters or
 - return values, 38
 - with no return value, 24
 - with parameters, 25
 - returning string, 24
 - multiple skins for same control
 - type, 224
 - old-fashioned ASP, 3-4
 - overloaded methods in Car
 - class, 28
 - overriding Render() method
 - in control, 148
 - page calling Web service, 179
 - Post class in user interface
 - glue, 58
 - private DeleteCache()
 - method, 82
 - private PopulateDefault()
 - method, 70
 - ProfileInfo with
 - ProfileInfoCollection, 219
 - programmatically wiring up
 - events in page, 114
 - properties in derived
 - MembershipProvider, 193
 - property and matching private
 - variable, 22
 - proxy class file generated for
 - EsteemBooster Web
 - service, 177
 - read-only property, 23
 - record-specific constructor,
 - 71-72
 - refactored class without
 - properties, 39-40
 - Render() method and HTML
 - rendered for browser, 161
 - rendered form tag in a Web
 - form, 108
 - required properties of Profile
 - provider, 211
 - retrieving data in ASP.NET, 4
 - revised constructor with
 - caching, 78-80
 - revised CustomerID
 - property, 86
 - SetPropertyValues() method,
 - 217-218
 - setting themes
 - with Profile, 226
 - programmatically, 222
 - setting web.config
 - to use custom providers, 201
 - to use SqlMembership
 - Provider, 189
 - skin controls, 224
 - SMTP exchange, 292
 - start of data class, 64-68
 - static method, 29
 - for ArrayList full of
 - Customer objects, 83-85
 - StreamReader class, 290
 - StreamWriter class, 290
 - string concatenation versus
 - StringBuilder, 257
 - test fixture, 271
 - traditional code versus
 - declarative code, 230
 - unlocking your car (object
 - inheritance), 10
 - Update() method, 75-76
 - UpdateUser() method of
 - derived
 - MembershipProvider, 199
 - user controls, using, 12
 - variable scope, 26
 - viewstate rendered to the
 - finished page, 106
 - Load event, 110
 - LoadPostData() method, 149
 - local files, Visual Studio, 243
 - Login control, 202
 - LoginName control, 202
 - LoginStatus control, 202
 - LoginView control, 131, 203
 - Longhorn, 227

M

 - machine.config file
 - (HttpHandlers), 117, 119
 - management
 - code testing and deployment,
 - 267-280
 - TDD, 268-278
 - Visual Source Safe, 279
 - xcopy deployment, 278
 - viewstate, 253-254
 - measuring time (Trace), 258
 - members
 - defined, 17
 - naming conventions, 21
 - scope, defined, 26-27
 - membership, 183-185, 203
 - authentication controls,
 - 202-203
 - built-in membership provider,
 - 186-189
 - custom providers, 189-195,
 - 198-201
 - profile system
 - custom profile providers,
 - 209-213, 216-220
 - SqlProfileProvider class,
 - 205-209
 - MembershipUser class, 185
 - MessageName attribute, 169

306 Index

methods

AddWithValue(), 51, 71
 car analogy, 7-9
 ChangePassword(), 185
 ChangePasswordQuestionAndAnswer(), 185
 Close(), 288
 Create(), 73-74
 CreateChildControls(), 156
 data access, 254-255
 defined, 7-8, 23-26
 Delete(), 76-77
 DeleteCache(), 82
 Dispose(), 130
 ExecuteNonQuery(), 76
 Get..., 142
 GetPassword(), 185
 GetProfile(), 212
 GetPropertyValues(), 212, 216
 GetResponse(), 295
 GetRoles(), 131
 GetUser(), 195
 Init(), 130
 Initialize(), 193, 209
 InitializeComponent(), 115
 LoadPostData(), 149
 Membership class static methods, 184
 OnInit, 115
 OnSelectedIndexChanged(), 144
 overloading, 27-28
 overriding, 10
 PopulateDefault(), 70
 PopulateMembershipUser(), 198
 post, 108
 ProcessRequest(), 124
 profile providers
 DeleteInactiveProfiles(), 219
 DeleteProfiles(), 219
 FindInactiveProfilesByUserName(), 221
 FindProfilesByUserName(), 221
 GetAllInactiveProfiles(), 221
 GetAllProfiles(), 220
 GetNumberOfInactiveProfiles(), 220
 Profile.Save(), 207
 protected, 162
 RaisePostDataChangedEvent(), 149-150

Read(), 295
 RebelThread(), 298
 Remove(), 82
 Render(), 147, 155
 RenderBeginTag(), 161
 RenderEndTag(), 161
 ResetPassword(), 185
 Roles class static methods, 185
 SendMessageButton_Click, 115
 SendRequest(), 295
 SetPropertyValues(), 212, 216-218
 Start(), 296
 static methods, 28-29, 41-42
 Stream class, 284
 StringBuilder, 255
 TimerMethod(), 300
 Trace.Warn, 258
 Trace.Write, 258
 Update(), 75-76
 UpdateUser(), 198
 versus properties, designing classes, 37-41
 versus subroutines, 24
 Write(), 295
 WriteEndElement(), 162
 WriteFile(), 124
 WriteLine(), 277
 WriteStartElement(), 162
 modifications, proxy class, 180
 modifiers. *See* access modifiers
 MSIL (Microsoft Intermediate Language), 95
 multiple projects, Visual Studio, 242-243

N

n-tier application architecture, 47-48
 Name property
 ProfileProvider, 209
 WebService attribute, 170
 Namespace property (WebService attribute), 170
 namespaces, declaring classes, 18
 NameValueCollection, 195
 naming conventions, 21, 269
 navigation controls, declarative programming, 233-234
 needs assessment. *See* requirements assessment

networking classes, 290-292, 295-296
 NUnit, 280
 test fixtures, 268
 nvarchar, 64

O

object parameters, 113
 object-oriented programming
 custom data classes. *See* custom data classes
 versus script-style programming, 3-6
 Web services, 180
 object/relational mapping, defined, 42
 objects. *See also* classes
 CacheDependency, 251
 car analogy, 7-9
 defined, 5-7
 inheritance, 9-13
 instantiating, 19
 with parameters, 20
 Principal, 131
 SettingsContext, 212
 SettingsPropertyCollection, 212
 SettingsPropertyValueCollection, 212
 Trace, 258, 265
 Web services as, 165, 181
 consumption, 173-174, 177-179
 converting code to, 166-173
 modification of proxy class, 180
 object-oriented programming, 180
 OnClick attribute, 114
 OnInit method, 115
 OnSelectedIndexChanged() method, 144
 onsubmit attribute, 108
 output parameters, 220
 OutputCache directive, 250
 overloading methods, 27-28
 overriding methods, 10

P
 page compilation, 95
 code-behind, compiled, 99
 code-behind, non-compiled, 97-99

- code-beside/partial classes, 100-103
 - inline code, 96-97
 - @Page directive, 127
 - enableViewState attribute, 254
 - page events, 111-113
 - page postback system, 107, 110
 - Page_Load event handler, 115
 - ParameterCollection class, 51
 - ParameterizedThreadStart
 - delegate, 298
 - parameters
 - EventArgs, 113
 - FileAccess, 285
 - FileMode, 285
 - instantiating objects, 20
 - object, 113
 - output, 220
 - passing
 - to constructors, 20, 41
 - to methods, 25
 - partial classes, 57
 - PasswordRecovery control, 203
 - performance, 249, 264
 - data access methods, 254-255
 - data caching, 250-253
 - fragment caching, 250
 - hardware, 261
 - measuring time with Trace, 258
 - Performance monitor, 259-260, 265
 - Query Analyzer, 261, 264-265
 - SQL Profiler, 261, 264-265
 - string concatenation versus
 - StringBuilder, 255-257
 - viewstate management, 253-254
 - Performance monitor, 259-260, 265
 - Performance tab, 260
 - pollTime attribute, 252
 - POP Forums architecture case study, 48-59
 - PopulateDefault() method, 70
 - PopulateMembershipUser()
 - helper method, 198
 - post method, 108
 - postback events, 113-114
 - postback process, 107-110
 - pre-compiled code deployment, 103-104
 - PreRender event, 110
 - Principal objects, 131
 - private methods, defined, 26
 - private variables, defined, 21-23
 - Processes tab, 260
 - ProcessRequest() method, 124
 - Profile property (HttpContext class), 205-208
 - profile providers, customization, 209-213, 216-220
 - profile system
 - custom profile providers, 209-213, 216-220
 - SqlProfileProvider class, 205-209
 - Profile.Save() method, 207
 - ProfileInfo class, 218
 - ProfileInfoCollection class, 219
 - ProfileProvider properties, 209
 - properties
 - AutoPostBack, 144
 - car analogy, 7-9
 - custom data classes, 64, 68-69
 - defined, 7-8, 21-23
 - EnableViewState, 254
 - Encoding, 288
 - HttpContext class, Profile, 205-208
 - HttpHandlers, 120-121
 - MembershipProvider, 191, 200
 - ProfileProvider, 209
 - Stream class, 283
 - versus methods, designing classes, 37-41
 - WebService attribute, 170
 - PropertyNames column, 213
 - protected methods, 162
 - provider model, 184
 - providers
 - built-in membership provider, 186-189
 - custom providers, 189-195, 198-201
 - proxy class, modification, 180
 - public methods, defined, 26
 - Publish option (Website menu), 278
- Q-R**
- Query Analyzer, 261, 264-265
 - question-and-answer challenge, Membership system, 185
 - RaisePostDataChangedEvent()
 - method, 149-150
 - RangeValidator control (declarative programming), 228
 - Read() method, 295
 - read-only properties, defined, 23
 - RebelThread() method, 298
 - record-specific constructor, 71-72
 - records (databases), retrieving, 3-4
 - references, Visual Studio, 243
 - Register directive, 141
 - RegularExpressionValidator
 - control (declarative programming), 228
 - remote servers, Visual Studio, 244
 - Remove() method, 82
 - Render() method, 147, 155
 - RenderBeginTag() method, 161
 - RenderEndTag() method, 161
 - rendering HTML, server
 - controls, 155
 - Repeater controls, 86
 - Request property (HttpHandlers), 120
 - requests, Web applications, 91
 - IIS, 91, 94-104
 - RequiredFieldValidator
 - control (declarative programming), 228-229
 - requirements assessment
 - application architecture, 60
 - designing classes, 35-36
 - ResetPassword() method, 185
 - Resources prefix (default expression), 235
 - Response property (HttpHandlers), 120
 - return value (methods), 24-25
 - role-based security, HttpModules, 130, 134, 137
- S**
- scalability, 249
 - scope, defined, 26-27
 - script-style programming
 - versus object-oriented programming, 3-6
 - SDK (software development kit), 240

308 Index

- security, forms authentication and membership, 183-185, 203
 - built-in membership provider, 186-189
 - controls, 202-203
 - custom providers, 189-195, 198-201
- sender object parameter, 113
- SendMessageButton_Click method, 115
- SendRequest() method, 295
- serialization, 165
- server controls, 139, 163
 - building controls, 145-151
 - composite controls, 156, 159-161
 - versus user controls, 162-163
 - derived controls, 139-145
 - events, 110-111
 - rendering HTML, 155
 - viewstate, 155
- server hardware, performance, 261
- service-oriented architecture (SOA), 59
- SetPropertyValues() method, 212, 216-218
- SettingsContext objects, 212
- SettingsPropertyCollection objects, 212
- SettingsPropertyValueCollection objects, 212
- shared methods. *See* static methods
- Simple Mail Transfer Protocol. *See* SMTP
- SiteMapDataSource control (declarative programming), 234
- SiteMapPath control (declarative programming), 233-234
- skin files, 223-225
- SkinID attribute, 224
- SMTP (Simple Mail Transfer Protocol), 290
- SOA (service-oriented architecture), 59
- sockets, 290
- software development kit (SDK), 240
- Solution Explorer, 269
- source control providers, Visual Studio, 245
- sprocs (stored procedures), 264
- SQL Profiler, 261, 264-265
- SqlCacheDependency class, 251
- SqlDataSource control (declarative programming), 229
- SqlMembershipProvider, 186-189
- SqlProfileProvider class (profile system), 205-209
- Start() method, 296
- static methods, 28-29
 - designing classes, 41-42
 - Membership class, 184
 - Roles class, 185
- stored procedures (sprocs), 264
- Stream class, 33
 - methods, 284
 - properties, 283
- StreamReader class, 288
- streams, 283-285
 - Stream class, 33
 - methods, 284
 - properties, 283
 - StreamReader class, 288
 - StreamWriter class, 288
- StreamWriter class, 288
- string concatenation, 265
 - versus StringBuilder, 255-257
- StringBuilder, 265
 - versus string concatenation, 255-257
- subroutines versus methods, 24
- System.EventHandler delegate, 113
- System.Object, inheritance from, 14-15
- T**
- tables
 - aspnet_Profile, 213
 - aspnet_Users, 213
 - control, page and application events, 112
 - Customers, 63
 - FileMode enumeration, 285
 - static methods
 - Membership class, 184
 - Roles class, 185
 - Stream class
 - methods, 284
 - properties, 283
 - tags, anonymousIdentification, 208
 - Task Manager, 260
 - TCP (Transmission Control Protocol), 290
 - TcpClient class, 290
 - TcpListener class, 290
 - TDD (test-driven development), 43-44, 267-281
 - Test attribute, 270
 - test fixtures (NUnit), 268
 - Test-Driven Development in Microsoft .NET*, 281
 - TestFixture attribute, 270
 - testing and deployment, 267-280
 - TDD, 268-278
 - Visual Source Safe, 279
 - xcopy deployment, 278
 - Theme attribute, 221
 - themes, 221-222
 - user profiles, 225-226
 - this keyword, 115
 - threads, 296-300
 - ThreadStart delegate, 298
 - tiers (application architecture), 60
 - defined, 45
 - n-tier architecture, 47-48
 - POP Forums architecture case study, 48-59
 - requirements assessment, 60
 - TimerCallback delegate, 300
 - TimerMethod() method, 300
 - ToString() method, 255
 - Trace object, 265
 - measuring time, 258
 - Trace.Warn() method, 258
 - Trace.Write() method, 258
 - TransactionOption attribute, 169
 - Transmission Control Protocol. *See* TCP
 - TreeView control (declarative programming), 233

U

UDP (User Datagram Protocol), 290
 UdpClient class, 290
 unicode, nvarchar, 64
 Unload event, 110
 Update() method, 75-76
 UpdateCommand attribute, 232
 UpdateUser() method, 198
 user controls
 inheritance, 12-13
 Label controls, 11
 versus composite controls, 162-163
 User Datagram Protocol. *See* UDP
 user identification, forms
 authentication and membership, 183-185, 203
 built-in membership provider, 186-189
 controls, 202-203
 custom providers, 189-195, 198-201
 user interface layer, connecting to business rules layer, 57-58
 user profiles, themes, 225-226

V

validation controls, declarative programming, 228-229
 ValidationSummary control (declarative programming), 228
 variables
 private variables, defined, 21-23
 scope, 26
 versions, Visual Studio, 239-240
 viewstate, 106
 management, 253-254
 server controls, 155
 Visual Source Safe, 279-281
 Visual Studio, 239, 247
 debugger, 246-247
 default behavior, 241
 local files, 243
 multiple projects, 242-243
 references, 243
 remote servers, 244
 source control providers, 245

versions, 239-240

Web form designer, 241-242

Web servers, 244-245

VWD (Visual Web Developer), 240

W-Z

WAS (Web Application Stress Tool), 258-259, 265

Web applications, 91, 94-104

Web form designer, Visual Studio, 241-242

Web Parts, declarative programming, 232

Web servers, Visual Studio, 244-245

Web Service Definition Language. *See* WSDL

Web services, 165, 181

 consumption, 173-174,

 177-179

 converting code to, 166-173

 modification of proxy class, 180

 object-oriented

 programming, 180

@WebHandler directive, 127

WebMatrix, 240

WebMethod attribute, 166

WebPartManager control (declarative programming), 232

WebService attribute,

 properties, 170

Website menu, Publish

 option, 278

wildcards, 94

Windows Forms

 Programming, 116

wiring events, 114-115

Write() method, 295

WriteEndElement()

 method, 162

WriteFile() method, 124

WriteLine() method, 277

WriteStartElement()

 method, 162

WSDL (Web Service Definition

 Language), 177

wsdl.exe program, 178

XAML (Extensible Application

 Markup Language), 227

xcopy deployment, 278

XmlTextWriter class, 162