

Symbols

!= 82
 with generics 76
#pragma 197
*. *See* transparent identifiers
:: 194
== 82
 with generics 76
=> 233
? modifier 121
?? operator. *See* null coalescing operator

A

about:blank, as URL equivalent of null
 reference 113
abstract 175
 implicit for static classes 191
abstract classes 256, 271
abstraction 68
academic purity 220
access modifiers
 default 193
 for partial types 186
 for properties 192
Action 12, 97, 145, 232, 236
Active Directory 347
Active Server Pages (ASP) 19
Adapter 98
adapters 336
Add 219, 286
addition using expression trees 239
ADO.NET 114, 319, 334–336
 Entity Framework 22, 315, 348
Adobe 22, 24

aesthetics 222
Aggregate. *See* Standard Query Operators,
 Aggregate
AJAX 24
aliases 213
aliases. *See* namespace aliases
All. *See* Standard Query Operators, All
Amazon 344
 web service 17
Ancestors 342
AncestorsAndSelf 342
angle brackets 68
Annotations method 342
anonymous functions 232, 249, 253
 conversions 252
anonymous methods 10, 55, 144–159, 245,
 265, 277, 356
 ambiguity 150
 compared with lambda expressions 232
 implicit typing 212
 lifetime 159
 parameters
 ignoring 149
 using 145
 returning from methods 154
 returning values 147
anonymous object initializers 57, 224
anonymous types 57, 224, 236, 266, 340
 members 226
 number of distinct types 225
 transparent identifiers 296
Any. *See* Standard Query Operators, Any
Append 265
applet 18
application domains 49
Application.Run 139

architecture 275
 AreaComparer 105
 argument evaluation 189
 argument, parameter passing 53
 arithmetic 239
 Array 43
 array indexing 239
 ArrayList 5, 8, 46, 52
 comparisons with List<T> 65, 73, 88, 96
 use with LINQ 289
 arrays 278, 360
 as reference types 49
 covariance 45
 covariance support 103
 implicit typing. *See* implicitly typed arrays
 initialization 223
 ASCII 44
 AsEnumerable, DataTable extension method 335
 See also Standard Query Operators,
 AsEnumerable
 ASP Classic 20
 See also Active Server Pages (ASP)
 ASP.NET 20, 26, 198
 AsParallel 349
 AsQueryable. *See* Standard Query Operators,
 AsQueryable
 AsReadOnly 98
 assemblies 225, 261
 extern aliases 196
 assembly reference 28
 AssemblyInfo.cs 203
 assignment 50–51
 of null to a nullable type variable 120
 assignments 242
 associations 316, 337
 asterisk 297
 AsXAttributes 340
 asynchronous code 178
 asynchronous pattern 81
 attorney 34, 38
 Attribute method 342
 Attribute suffix 201
 attributes 316
 Attributes method 342
 automatically implemented properties
 208–210, 235
 Average. *See* Standard Query Operators, Average
 axis methods 342
 Ayende 345

B

back tick 93
 BackgroundWorker 41
 backward compatibility 253
 Base Class Libraries (BCL) 27

base class library 207
 base type 191
 specifying for partial types 185
 BCL. *See* Base Class Libraries (BCL)
 BeginXXX/EndXXX pattern 81
 behavior
 adding using inheritance 255
 specifying with delegates 33
 behavioral pattern 161
 best practices 209, 274, 356
 better conversion 251
 Bill Gates 31
 binary operators 125
 BinaryExpression 239
 black box view of iterator blocks 167
 bloat of type responsibility 187
 block
 anonymous methods 146
 lambda expressions 234
 restrictions of expression trees 242
 blogs 270, 272, 315, 345
 blueprints 68
 Boolean
 flags 115
 logic 127
 bounds of type variables 248
 boxing 53–54, 58, 89–90, 115, 119, 178
 in Java 110
 in Java 1.5 20
 of Nullable. *See* Nullable, boxing and unboxing
 braces 147, 234
 breaking changes from C# 1 to C# 2 144
 buffer 257
 buffering 265, 281, 308
 bugs
 due to lack of generics 65
 using Stream 257
 bulk delete 319
 business
 code 174
 layer 348
 logic 187
 requirements 269
 rules 130
 Button 139, 193
 byte, range of values 113
 bytecode 110
 ByValArray 201
 by-value argument. *See* parameter passing

C

C ∅ 21, 24, 33, 44
 C#
 definition of events 40
 evolution 4

- C# (*continued*)
 - influences 18
 - language distinction 25
 - language specification 117
 - version numbers 26
- C# 1
 - comparison of delegate usage with C# 3 233
 - delegate creation expressions 138
 - event handler plumbing 138
 - iterator implementation 162
 - patterns for null values 114
 - revision of key topics 32–54
- C# 2
 - delegate improvements 137–160
 - fixed size buffers 199
 - generics 63–111
 - iterators 161–182
 - namespace aliases 193
 - nullable types 112–136
 - partial types 184
 - pragma directives 197
 - separate property access modifiers 192
 - static classes 190
 - support for nullable types 120
 - terminology for nullable types 117, 120
- C# 3
 - anonymous types 224
 - automatically implemented properties 208–210
 - collection initializers 218–221
 - expression trees 238–245
 - extension methods 255–274
 - implicitly typed arrays 223
 - implicitly typed local variables 210, 215
 - lambda expressions 230
 - object initializers 215–218
 - overloading 251–253
 - partial methods 188
 - query expressions 275–313
 - two-phase type inference 248
 - type inference 245–253
- C# 4 106, 358
- C# team 48
- C++ 20, 24, 83, 102, 108, 201
- C++/CLI 24
- C++0x 108
- caching of delegate instances 237
- caller, parameter passing 52
- captured variables 150–159, 285
 - extended lifetime 154
 - instantiations 155
- Cartesian product 304
- cascading insertion 319
- cast 45
- Cast. *See* Standard Query Operators, Cast
- casting
 - anonymous methods 150
 - as a necessary evil 9, 64
 - migrating to generics 74
 - necessity removed by generics 67, 178
 - range variables 290
 - reference identity 51
 - type inference 80, 223, 247
 - unsafe abuse in C 44
 - untyped datasets 334–335
- Catchphrase 271
- CCR. *See* Concurrency and Coordination Runtime (CCR)
- chaining
 - extension methods 265–266, 354
 - iterators 265
- change tracking 318, 348
- characters 278
- checksum pragmas 198
- clarity 272
- clashes of names 194
- class declaration, absence of in snippets 28
- class libraries 255
- class. *See* reference types
- classes as generic types 67
- classification of variables 151
- CLI. *See* Common Language Infrastructure (CLI)
- Click 139, 142, 149
- Clone 46, 98
- cloning 226
- closed types 69, 86
- closures 145, 151, 160
- CLR. *See* Common Language Runtime (CLR)
- code
 - as data 238
 - bloat 108
 - generation 115
 - generators 184, 186, 188
 - shape 222
 - smells 136
- CodeDOM 239, 241, 244
- coding
 - conventions 209
 - standards 182
 - style, harmony with language 4
- collection initializers 8, 218–221, 235, 286, 355
- CollectionBase 46
- collections 56
 - generic 66
 - generic collections in .NET 2.0 96
 - removing multiple elements 98
 - specialized 46
 - strongly typed vs weakly typed 45

- COM 211
 - See also* Component Object Model (COM)
- COM3 19
- Combine 38, 138
- command line 196
- commas, separating type parameters 68
- comments 158
- Common Language Infrastructure
 - annotated standard 103
 - designers 103
 - specification 77, 120
- Common Language Infrastructure (CLI) 19, 25
- Common Language Runtime (CLR)
 - anonymous types 226
 - aspects of generics 85
 - described 25
 - fixed buffer allocation 199
 - generics 88
 - lack of change for .NET 3.5 207
 - modifications for boxing of Nullable 118
 - multi-language support 354
 - nullable types 128
 - type safety 45
 - version numbers 26
- community 184, 213, 272
 - influence on nullable type behavior 118
- Community Technology Preview (CTP)
 - Parallel Extensions 349
- compact code 146, 232, 253
- compactness 146
- Compare 133
- Comparer<T> 83
- Comparer<T>.Default 134
- CompareTo 82
- Comparison<T> 9, 138, 148, 236, 356
- comparisons 239
 - in generics 82
 - using the null coalescing operator 133
 - with null, unintended 126
- compatibility
 - inferred type parameters 249, 253
 - of delegate signatures 55
- compatible declarations for partial types 185
- competition between technologies 22
- compilation errors 288, 294
- Compile method of LambdaExpression 241, 362
- compiled code 211
- compiler 25
 - application of Attribute suffix 201
 - applying complex transformations 181
 - captured variable implementation 158
 - checking delegate return value types 147
 - checking value type mutation 116
 - choices in overloading 251
 - code validation 111
 - combining files for partial types 185
 - creating extra classes for captured variables 155
 - creating state machine for iterators 166
 - creation of expression trees 244
 - generating anonymous methods 146
 - giving information with casts 53, 64
 - handling of null 122
 - handling of query expressions 276
 - implementation-specific features 197
 - inferring array types 223
 - inferring delegate types 139
 - inferring local variable types 57, 211
 - knowledge of framework 350
 - knowledge of static classes 191
 - lambda expression expansion 237
 - logic for type inference 80, 248
 - preventing assignment of null to normal value types 113
 - preventing definitely invalid casts 45
 - quirks in iterator blocks 172
 - removing degenerate query expressions 292
 - responsibility for language improvements 54, 208
 - timing of lambda expression body checking 251
 - transforming delegate shorthand 39
 - translations of query expressions 284
 - warnings 197
- compile-time
 - checking 65, 244, 325, 350
 - knowledge of variable type 44
 - lack of type safety with weakly typed collections 46
 - overloading 83
 - safety 245
 - type 211–212, 223
- complete programs 28
- complex numbers 107
- complexity 145, 159, 184, 244, 314, 348
 - in appropriate place 174
 - of language 48
- Component Object Model (COM) 47
- computer science 231
- Concat. *See* Standard Query Operators, Concat
- concatenation. *See* Standard Query Operators, concatenation
- concepts in C++ templates 108
- conceptual model 348
- concurrency 324, 357
- Concurrency and Coordination Runtime (CCR) 178, 358
- conditional logical operators 127
- conditional operator 121, 129
- configuration files 187
- conflicts 324

- confusion 150, 222
 - over captured variables 157
 - connection management 318
 - connection string 316, 336
 - consistency
 - code formatting 147, 149
 - language design 253
 - of immutable types 39
 - of inferred type arguments 80
 - of nullable equality rules 119
 - of nullable syntax choice 121
 - of querying with LINQ 281, 325, 337, 344, 350
 - Console 67, 194, 237, 320
 - Constant 329
 - constant collections 221
 - constants 212
 - in expression trees 243
 - constraint violations 319
 - constraints on type parameters. *See* type constraints
 - constructed types 68, 92
 - constructor parameters, anonymous types 226
 - constructor type constraints 76
 - constructors 108, 191
 - generated 188
 - in generic types 70
 - of `Nullable<T>` 116
 - parameterless 77, 210, 217
 - private in utility classes 190
 - signatures other than parameterless 77
 - static. *See* static constructors
 - `XElement` 339
 - Contains. *See* Standard Query Operators, Contains
 - ContainsKey 99
 - context 151, 271, 287
 - contextual keywords 298, 310
 - partial 185
 - continuation-passing style 180
 - contracts in Spec# 21
 - contravariance 103, 105
 - message group conversions 141
 - of delegates 55
 - supported by CLR 106
 - conventions 193, 209, 271, 273
 - for events 142
 - conversion
 - of lists using delegates 72
 - standard query operators. *See* Standard Query Operators, conversion
 - conversions
 - from method group to delegate instance 55
 - preference in overloading 251
 - `ConvertAll` 71, 96, 227, 267, 277, 281
 - `Converter` 72, 175
 - conveyor belt 278
 - `COOL` 19
 - coordination 178, 350
 - `Copy` 257
 - copy behavior for value/reference types 49
 - `CopyToDataTable` 335
 - corner cases 85
 - coroutine 182
 - correspondence between queries on different data sources 16
 - `Count` 98
 - `Count`. *See* Standard Query Operators, `Count`
 - covariance 103
 - method group conversions 141
 - of arrays 45
 - of delegates 55
 - of return type 46
 - supported by the CLR 106
 - `CPU` 88
 - `CreateQuery` 327, 330
 - criteria for matching 97
 - cross joins 303
 - CTP. *See* Community Technology Preview (CTP)
 - `Current` 90, 162, 164, 167, 172, 279
- ## D
-
- data
 - contexts 17, 318
 - models 276
 - pipeline 161
 - source 14
 - sources 276
 - transfer 51
 - transfer type 57
 - Data Access Strategy 348
 - data binding 283
 - Data Connections 316
 - data models 276
 - `DataAdapter` 319
 - databases 179, 269, 278, 313–314
 - access 131
 - column selection 228
 - independence 346
 - joins 297
 - nullable fields 58, 114
 - recreating 319
 - data-centric approach 11
 - `DataReader` 265, 278
 - `DataRow` 335
 - `DataRowExtensions` 334
 - `DataSet` 265
 - datasets
 - typed 335–338
 - untyped 334–335
 - `DataTable` 334–335

- DataTableExtensions 334–335
- DateTime 51, 82, 116, 125, 283
 - non-nullability 113
- DateTimeRange 301
- DBNull 114, 334–335
- debugger 198, 321
- declaration
 - of local variables 211, 213
 - of partial methods 189
- declarative 14
- declarative programming 313
- decompilation 123, 225
- deduction event handlers 139
- default
 - access modifier 193
 - mode of parameter passing 54
- default constructor 191
 - of Nullable<T> 116, 120, 122
- Default property of EqualityComparer<T> and Comparer<T> 83
- default value expressions 81
- default values 172, 210
 - for type parameters 75*See also* default value expressions
- DefaultIfEmpty. *See* Standard Query Operators, DefaultIfEmpty
- deferred execution 264, 267, 279, 290, 303, 312, 337
- definite assignment 210
 - output parameters 81
 - variables 65
- degenerate query expressions 291
- delegate creation expression 143
- delegate instance 34–38, 284
 - always referring to a method 35
 - clumsiness of C# 1 54
 - created by anonymous methods 146
- delegate keyword 146, 149
 - for anonymous methods 10
- delegate type inference 227
- delegate types 34
 - compatibility 143
 - declaration 34, 37
 - inferred by the compiler 139
- delegate, used for sorting 9
- delegates 354
 - action 35, 140, 142
 - as generic types 67
 - as reference types 49
 - C# 2 improvements 137–160
 - C# invocation shorthand 36
 - candidate signatures 35
 - combining 38–40
 - compiling from expression trees 240–241
 - contravariance 35, 55, 138
 - covariance 138
 - creating instances 34
 - creating one instance from another 143
 - events 40–41
 - examples of combination and removal 39
 - exceptions 40
 - function pointer 33
 - immutability 38
 - in C# 1 33–42
 - in the framework 138, 145, 153, 160
 - instances. *See* delegate instances
 - invocation 34, 36–38, 138, 154
 - Invoke method 36
 - lambda expressions 232–238
 - method group conversions 73
 - specifying behavior 33
 - summary of C# 1 features 41
 - target 140
 - target of action 36
 - types. *See* delegate types
 - used in List 96
 - used in the CCR 181
 - using for specialization 355
- DELETE 319
- Dequeue 100
- derivation type constraints 77
 - example with default values 81
 - restrictions 78
- DescendantNodes 342
- DescendantNodesAndSelf 342
- Descendants 342
- DescendantsAndSelf 342
- descending ordering 294
- design layers 353
- design overhead 255
- design patterns 161, 181, 222, 274
- designers 186
 - DataSets 335
 - LINQ to SQL 316, 319
 - Windows Forms 184
- developers 353
- development platform 26
- diagrams, LINQ to SQL 316
- dictionaries 219
- Dictionary<TKey,TValue> 66, 99, 211
- Dictionary`2<TKey,TValue> 93
- DictionaryEntry 67
- directories, LDAP 347
- disabling warnings 198
- disassembly 123
- discretion 208
- discussion groups 113
- Dispose 25, 92, 168, 171, 193
- Distinct. *See* Standard Query Operators, Distinct
- DLR. *See* Dynamic Language Runtime (DLR)

- Document Object Model (DOM) 338–339, 343
 - documentation 273
 - DOM. *See* Document Object Model (DOM)
 - domain specific languages 271
 - doubly linked lists 101
 - DSL. *See* domain specific languages
 - duck typing, relationship to query expressions 286
 - Dyer, Wes 270
 - dynamic configuration 92
 - Dynamic Language Runtime (DLR) 22, 355
 - dynamic languages 24, 211, 354
 - dynamic typing 42–43, 48, 65, 211, 354
- E**
-
- early out, of iterator blocks 169
 - ECMA 19, 85
 - e-commerce application 113
 - Effective Java 85
 - efficiency 265–266, 280
 - of value types 54
 - elegance 136
 - of static classes 190
 - Element method 342
 - element operations. *See* Standard Query Operators, element operations
 - ElementAt. *See* Standard Query Operators, ElementAt
 - ElementAtOrDefault. *See* Standard Query Operators, ElementAtOrDefault
 - Elements method 342
 - ElementsAfterSelf 342
 - ElementsBeforeSelf 342
 - ElementType 326
 - ellipsis, in snippets 29
 - embedded objects 218
 - emphasis 133
 - empty strings 50, 67
 - Empty. *See* Standard Query Operators, Empty
 - enabling warnings 198
 - encapsulation 178, 184, 192, 216, 222, 267
 - events 41
 - of behavior in delegates 33
 - of iteration 174
 - of the iterator pattern 161
 - of variables within properties 6
 - sequences 278
 - enhanced for loop 20
 - Enqueue 100
 - entities 187, 316
 - Entity Client 348
 - Entity Framework. *See* ADO.NET Entity Framework
 - entity inheritance strategies 348
 - Entity SQL 348
 - Enum 342
 - enum. *See* enumerations
 - Enumerable 263–277, 314, 326, 364
 - enumeration
 - and iteration 162
 - of a linked list 102
 - of generic collections 86
 - enumerations 316, 335
 - as value types 49
 - enums 20, 271
 - environment context of a closure 151
 - equality operators 119, 125
 - on nullable types 125
 - EqualityComparer<T> 83
 - Equals 53, 82–83, 99, 133, 226
 - equals, contextual keyword 298
 - equijoins 303
 - erasure, transparent identifiers 297
 - Error List window 198
 - error, during type inference 248
 - errors 150
 - eSQL. *See* Entity SQL
 - Evans, Eric 271
 - event handlers 36, 138, 142, 148, 235, 237, 336
 - See also* events
 - EventArgs 141–142, 186, 238
 - EventHandler 55, 138, 141–142
 - events 40–41, 108, 188, 235, 318
 - logging 141
 - no-op handlers 149
 - subscribing with method group conversions 140
 - Except. *See* Standard Query Operators, Except
 - exceptions 81, 262, 273, 361
 - thrown by delegate actions 40
 - thrown by iterators 173
 - thrown by nullable conversions 118
 - thrown for invalid casts 51
 - Exchange Server 2007 21
 - exclusive OR operator for bool? 127
 - exclusive ranges 175
 - Execute 327, 329, 332
 - ExecuteCommand 319
 - execution
 - model 244
 - of static constructors 87
 - plan 322
 - executive summary of nullable type conversions
 - and operators 124
 - Exists 96
 - exit points 148
 - experimentation 66
 - explicit conversions, XAttribute 342
 - explicit interface implementation 46, 90, 177
 - explicit type arguments 79

- explicit typing 43, 214, 354
 - lambda expression parameters 234
 - range variables 289
- Express editions of Visual Studio 30
- Expression 239–240, 326
- expression to initialize an object 217
- expression trees 238–245, 284, 325–326, 350, 358
 - combining with `IQueryProvider` 327
 - compilation 241
 - parameters for Queryable methods 330
- ExpressionType 239
- expressive 3
- expressiveness 111, 122, 154, 162
 - of C# 3 355
- extended type 258
- extending the world 270
- Extensible Application Markup Language (XAML) 187
- extension methods 10, 57, 255–274
 - calling 259
 - compiler discovery 261
 - declaration 258
 - Enumerable 285, 354
 - extended type 258
 - guidelines 270
 - LINQ to XML 342
 - Queryable 325–326
 - restrictions 258
- ExtensionAttribute 261
- extern aliases 194, 196

F

- F# 354
- factory methods 239
- FakeQuery 328
- FakeQueryProvider 328
- false operator 125
- fetching strategies 348
- Field 335
- field-like events 40
- fields 216
 - backing anonymous type properties 226
- FIFO. *See* first in first out
- files, spreading a type across multiple 185
- FileStream 257
- films 236
- filter expressions 291
- filtering
 - in SQL 321
 - LINQ to Objects 265, 279, 290
 - LINQ to XML 342
 - lists 15, 236, 356
- Filtering. *See* Standard Query Operators, filtering
- filters 161
- finalizers 108
- finally 92
- finally blocks
 - in iterator blocks 170–172
 - restrictions in iterator blocks 166
- FindAll 96, 153, 236, 356
- FindFirst 96
- FindLast 96
- first in first out 100
- First. *See* Standard Query Operators, First
- FirstOrDefault. *See* Standard Query Operators, FirstOrDefault
- fixed 199
- fixed type variables 248
- FixedBufferAttribute 199
- FixedSize 98
- fixing of type parameters 250
- flag
 - for faking nullity 13
 - to distinguish null values 115
- Flex 22
- flexibility 175, 281, 344, 348
- Flickr 344
- flow of iterator blocks 167
- fluent interfaces 271
- fluff 12, 28, 55, 219
- for loop 157, 159, 162, 174
 - implicit typing 213
- ForEach 146, 236
- foreach
 - and anonymous types 225
 - and captured variables 159
 - deferred execution 303
 - implicit casting 9
 - implicit typing 213
 - iterator pattern 162
 - nested collections 301
 - over generic collections 90
 - sequence diagrams 168, 280
- foreach loop 170, 173
- ForEach method 97
- foreach statement 181
- foreign keys 300, 316
- formatting
 - code 149
 - LINQ to XML source code 339
 - using delegates 355
 - within `DataTable.Select` 334
- forward references 185
- Fowler, Martin 271
- framework 128, 352
- framework libraries 25, 207
 - absence of range type 174
- frequency counting 66
- friend assemblies 201–204
- from clauses. *See* query expressions, from clause
- frustration with C# 1 144

fully qualified type name 197
 Func 56, 232, 288
 function 151
 functional 136, 160
 coding style 98, 244
 languages 231
 programming 235, 313, 353

G

.g.cs file extension for generated code 187
 garbage collection
 anonymous methods 155, 159
 fixed-size buffers 201
 iterator blocks 173
 language requirements 25
 of delegate instances 36
 value types 51, 115
 garbage collector 88
 performance penalty for excessive boxing 54
 generation. *See* Standard Query Operators,
 generation
 generic collections 289
 generic delegate types 72
 generic methods 67, 71, 74, 79, 106, 258
 in non-generic types 73
 generic type definition 92, 94
 generic type inference used with
 System.Nullable 120
 generic types 67
 nesting 86
 restriction on extension methods 258
 generics 56, 58–111
 benefits vs complexity 85
 delegate types 145
 experimenting with 74
 implementation 85
 implementing your own 81
 in Java 20
 lack of contravariance 103
 lack of covariance 103
 lack of generic properties and indexers 108
 limitations 102
 native code sharing 88
 nooks and crannies 85
 overloading by type parameter numbers 70
 performance benefits 88
 reading generic declarations 71
 type declaration syntax 69
 used for anonymous types 225
 GetConsoleScreenBufferEx 200
 GetEnumerator 90, 162–163, 169, 327, 332
 GetGenericArguments 94
 GetGenericTypeDefinition 94
 GetHashCode 53, 83, 99, 226
 GetMethods 95
 GetNextValue 175
 GetRange 98
 getter 192
 GetType 94
 GetValueOrDefault method of Nullable. *See*
 Nullable, GetValueOrDefault method
 global namespace alias 194–195
 golden rule of captured variables 159
 grammar 271
 Groovy 22, 48, 272, 355
 group ... by clauses. *See* query expressions, group
 ... by clauses
 group joins 301, 306
 GroupBy. *See* Standard Query Operators, GroupBy
 grouping 269, 307
 See also Standard Query Operators, grouping
 GroupJoin 333
 See also Standard Query Operators, GroupJoin
 GTK# 26
 guesswork 66
 GUID 199
 guidelines
 for captured variables 158
 for events 55, 142
 Gunnerson, Eric 107
 gut feelings 214, 273
 Guthrie, Scott 321

H

handwritten code 190, 226
 hash code, anonymous types 226
 hash codes 85
 HashSet<T>, in Java 110
 Hashtable 46, 66–67, 99, 131
 HasValue 13
 See also Nullable, HasValue property
 heap 51–52, 54, 155, 201
 allocation of wrappers 115
 Hejlsberg, Anders 107
 helper class for partial comparisons 134
 helper methods 190
 Hewlett-Packard 19
 Hibernate Query Language (HQL) 345
 Hibernate 22, 345
 hierarchy 222
 higher-order functions 235
 history 18
 hooks 188
 Hotspot 18
 HQL. *See* Hibernate Query Language (HQL)
 HTML 24
 Human Computer Interfaces 358
 human nature 353
 hyperbole 275

I

-
- IBM 20
 - ICloneable 46, 100
 - ICollection 220
 - IComparable 8, 53, 77–78, 82, 175
 - IComparer 8–9, 85, 99, 105, 138, 356
 - icon, for extension methods 260
 - IDE. *See* Integrated Development Environment (IDE)
 - identifiers 283, 287
 - identity management 318, 348
 - idioms 131, 233, 277
 - changes regarding delegates 54
 - IDisposable 25, 92, 162, 223
 - IEnumerable 335
 - IEnumerable<T>
 - collection initializers 219
 - explicit interface implementation 90, 177
 - extension methods 58, 263, 277–278, 286
 - generic form 70, 90
 - iterator blocks 165, 168
 - iterator pattern 161
 - LINQ and non-generic iterators 290
 - parallelization 349
 - relation to IQueryable 326, 330
 - specified behavior 172
 - IEnumerator 90, 161, 168, 171–172
 - IEqualityComparer 83, 99, 361, 363
 - IEquatable 82, 85
 - IGrouping<TKey,TElement> 270, 307, 365
 - IHashCodeProvider 99
 - IL. *See* Intermediate Language (IL)
 - ildasm 123, 128, 146, 172
 - IList 46, 96, 145, 256
 - ILookup 361
 - immediate execution 281, 303
 - immutability 240
 - anonymous types 226
 - of delegates 38
 - of Nullable 116
 - of ranges 177
 - impedance mismatch 276, 351
 - imperative programming 313
 - implementation 346
 - of captured variables 158
 - of partial methods 189
 - implicit
 - conversion from method groups to delegates 140
 - conversion operators 115
 - conversions 230
 - type arguments 79
 - implicit conversion
 - overloading 251
 - string to XName 338
 - type inference 248
 - implicit typing 43, 57, 219, 354
 - arrays 223, 246, 248
 - guidelines 213
 - local variables 210, 215, 235, 267
 - parameter lists 234, 246
 - restrictions 212
 - Visual Basic 47
 - inclusive OR operator for bool? 127
 - inclusive ranges 175
 - Increment 210
 - indentation 222
 - indexer 99, 102, 108
 - separate access modifiers 193
 - IndexOfKey 101
 - IndexOfValue 101
 - indirection 33, 38, 139
 - inferred return types 247, 253
 - role in overloading 252
 - information loss 114
 - inheritance 184, 255, 355
 - prohibited for value types 51
 - initialization
 - of local variables 211, 213
 - ordering in partial types 186
 - initialization expression 211, 213
 - initobj 123
 - inline
 - code for delegate actions. *See* anonymous methods
 - filtering 265
 - initialization 221
 - inlining 109
 - in-memory list 281
 - inner joins 297, 306
 - inner sequences 297
 - INSERT 319
 - InsertAllOnSubmit 319
 - instance variables created in iterators from local variables 167
 - instantiation of captured variables 155
 - instincts 273
 - int 52
 - integer literals 214
 - Integrated Development Environment (IDE) 111
 - Intel 19
 - intellectual curiosity 358
 - Intellisense 47, 64, 260, 263, 325
 - interfaces 90, 177, 191, 220, 226, 255, 285, 356
 - and extension methods 58
 - as generic types 67
 - as reference types 49
 - explicit implementation 46
 - implemented by partial types 185

- interfaces (*continued*)
 - no covariant return types 46
 - specifying multiple in derivation type constraints 78
 - Interlocked 210
 - Intermediate Language (IL) 25
 - code as data 238
 - compiled forms of LINQ queries 277
 - disassembling to discover implementation details 123, 146, 156, 172, 189, 225
 - generic type names 93
 - instruction to obtain MethodInfo directly 244
 - relation to Java bytecode 110
 - support for non-virtual calls 262
 - when using ? modifier 121
 - internal access modifier 201
 - InternalsVisibleTo 201–204
 - interoperability 49, 358
 - interpreted 18
 - Intersect. *See* Standard Query Operators, Intersect
 - into. *See* query expressions, join ... into and continuations
 - intuition 138
 - InvalidCastException 53
 - InvalidOperationException 116, 124
 - invariance 103
 - Inversion of Control 92
 - invocation of delegates 36–38
 - Invoke 36, 239
 - IEnumerable 349
 - IQueryable 326–328, 333
 - IQueryable<T> 261, 263, 286, 326–333, 360
 - IQueryProvider 326–328, 333
 - IronPython 22, 24, 355
 - IronRuby 355
 - IsGenericMethod 95
 - IsGenericType 93–94
 - IsGenericTypeDefinition 94
 - IShape 105
 - IsInterned 245
 - IsNullOrEmpty 262
 - ITask 180
 - iterable 162
 - iteration
 - manual vs foreach 172
 - over generic collections 90
 - iterator blocks 165–173, 177, 179, 258, 358
 - Microsoft implementation 172
 - sequence of events 168
 - using method parameters 169
 - yield type 166
 - iterator pattern 161
 - iterators 161–182
 - chaining together 265
 - implementing in C# 1 162
 - iterator blocks. *See* iterator blocks
 - restarting within iterator block 167
 - yield statements. *See* yield statements
-
- ## J
-
- J++ 19
 - jargon 68
 - Java 102, 190, 345, 355
 - 1.5 features 20
 - influence on .NET 22
 - influence over .NET 18
 - Java Virtual Machine (JVM) 19
 - Javascript 22
 - JavaServer Pages 18
 - JIT. *See* Just-In-Time (JIT) compiler
 - join ... into clauses. *See* query expressions, join ... into clauses
 - join clauses. *See* query expressions, join clauses
 - Join. *See* Standard Query Operators, Join
 - joins 15, 297
 - LINQ to SQL 322
 - LINQ to XML 343
 - standard query operators. *See* Standard Query Operators, joins
 - Just-In-Time (JIT) compiler 18, 25, 201
 - behavior with generics 65, 82, 88, 107
 - treating code as data 238
 - JVM. *See* Java Virtual Machine (JVM)
-
- ## K
-
- Key
 - property of DictionaryEntry 67
 - property of IGrouping 307
 - key
 - of grouping 307
 - of joins 297
 - key selectors 297
 - key/value pair 101
 - KeyPress 139
 - KeyPressEventArgs 141
 - KeyPressEventHandler 140–141
 - Keys 101
 - keys, in dictionaries 84
 - keystrokes 214
 - KeyValuePair<TKey,TValue> 67, 108, 220
-
- ## L
-
- lambda calculus 231
 - lambda expressions 10, 56, 137, 230–254, 277
 - body 234, 246, 251
 - built from query expressions 284–285, 288, 290, 303, 330
 - conversion to expression trees 241

- lambda expressions (*continued*)
 - in fluent interfaces 271
 - readability 356
 - role in LINQ 325
 - second phase of type inference 249
 - simplicity 12
 - use when filtering 265
- Lambda method of Expression class 240
- LambdaExpression 240, 362
- language changes 208
- language designers 113, 128, 136, 183, 204, 253
- Language INTe grated Query (LINQ) 14, 21, 56–57, 288–352
 - C# 2 as stepping stone 160–161
 - core concepts 275–281
 - deferred execution 279
 - functional style 354
 - joins 297
 - Standard Query Operators. *See* Standard Query Operators
 - type inference 251
 - use of anonymous types 224
 - use of expression trees 239, 244
 - use of extension methods 256, 263, 265
 - use of lambda expressions 231, 235
 - use of projection initializers 226
- language specification
 - collection initializers 220
 - embedded objects 218
 - extended type terminology 258
 - generics 85
 - iterator blocks 172
 - method group conversions 141
 - overload resolution 251
 - pragmas 197
 - query expression pattern 285
 - transparent identifiers 295, 297
 - type inference 248
- language-lawyer 80
- languages
 - adding functionality on nullable types 119
 - different behavior for nullable types 128
- last in first out 100
- Last. *See* Standard Query Operators, Last
- LastOrDefault. *See* Standard Query Operators, LastOrDefault
- laziness 214
- lazy code generation 88
- lazy loading 325, 337
- LDAP. *See* Lightweight Directory Access Protocol
- leaf nodes 240
- leaky abstraction 346
- learning 358
- left outer joins 301, 323
- Length 295
- LessThan 239
- let clauses. *See* query expressions, let clauses
- Libraries 25
- library changes 208
- library design 81
- lifetime of captured variables 154
- LIFO last in first out 100
- lifted conversions 124
- lifted operators 125, 128
- Lightning 19
- Lightweight Directory Access Protocol 347
- limitations of generics 102
- line breaks 146
- LinkedList<T> 101
- LINQ in Action 18, 315, 344
- LINQ providers 244, 276–277, 281, 292, 314, 333, 344
- LINQ. *See* Language INTe grated Query (LINQ)
- LINQ to Active Directory 347
- LINQ to Amazon 18, 344
- LINQ to DataSet 315, 334–338
- LINQ to Entities 348
- LINQ to NHibernate 345
- LINQ to Objects 275–313
 - lambda expressions 244
 - using with LINQ to DataSet 333–334, 337
 - using with LINQ to XML 343
- LINQ to SharePoint 347
- LINQ to SQL 17, 22, 314–326
 - compared with Entity Framework 348
 - debug visualizer 321
 - implicit joins 323
 - initial population 318–319
 - joins 322
 - model creation 315–318
 - queries 319–324
 - updating data 324–325
 - use of expression trees 244
 - use of IQueryable 330
- LINQ to XML 16, 315, 338–344
- LINQ. *See* Language INTe grated Query (LINQ)
- Lippert, Eric 106
- List<T>
 - collection initializers 219
 - comparison with ArrayList 88, 96
 - ConvertAll 71, 267, 277
 - Enumerable.ToList 360
 - FindAll 153
 - ForEach 146
 - lambda expressions 235–236
 - unstable sort 369
 - use of delegates in .NET 2.0 160, 356

- List`1 93
 - lists 278
 - local variables 151, 155, 285
 - declared in anonymous methods 146
 - implicit typing. *See* implicitly typed local variables
 - separate instances 155
 - stored in iterators 166
 - lock 209
 - lock statement 181
 - locking 192
 - log files 306
 - logging 188, 192, 228, 237, 318, 325, 332
 - logic 239
 - logic gates, in electronics 127
 - logical AND operator for bool? 127
 - logical model 348
 - LongCount. *See* Standard Query Operators, LongCount
 - looping in anonymous methods 146
 - loops
 - and anonymous methods 159
 - captured variables 156
 - in lambda expressions 234
 - lowest common denominator 32
- M**
-
- macros 20, 108
 - magic value
 - as substitute for null 114
 - for faking nullity 13
 - Main 28
 - mainstream 354
 - maintainability 159
 - maintenance 273
 - MakeGenericMethod 95
 - MakeGenericType 94
 - management 353
 - Manning 315, 344
 - mapping ADO.NET Entity Framework 348
 - Marguerie, Fabrice 344
 - marshaller 199
 - matching with predicates 236
 - Math 190
 - mathematical computing 106
 - Max. *See* Standard Query Operators, Max
 - maybe sense of null bool? value 128
 - mcs 199
 - MemberAccess 239
 - memory 50–52, 114
 - cost of linked lists 102
 - MemoryStream 51, 88, 142, 257
 - mental process 122
 - metadata 110, 315, 319
 - meta-provider 350
 - method arguments
 - combining for type inference 248
 - compared with type arguments 69
 - in expression trees 243
 - used in type inference 80
 - method calls
 - in expression trees 243
 - result used for implicit typing 212
 - method chaining 354
 - method exit points 169
 - method groups 140, 230, 244, 246
 - ambiguous conversions 141
 - converting into delegates 140
 - covariance and contravariance 141–144
 - implicit typing 212
 - second phase of type inference 249
 - method parameters
 - compared with type parameters 69
 - contravariance 105
 - used in iterator blocks 169
 - method signatures, reading generics 72
 - MethodInfo 95, 243
 - MethodInvoker 143
 - methods as value type members 51
 - metrics 269
 - micro-optimization 266
 - Microsoft
 - C# compiler implementation 158, 172, 197–198, 237
 - design guidelines 93, 95
 - language designers 120, 184, 358
 - language specification 85
 - LINQ 338, 344, 347–350
 - reasons for creating .NET 18, 20, 24, 31
 - Research 21, 354
 - Microsoft Robotics Studio 178
 - Min. *See* Standard Query Operators, Min
 - mindset 271
 - mindshare 20
 - mini-CLR 22
 - MinValue of DateTime 114
 - Miscellaneous Utility Library 174, 257, 349
 - MiscUtil. *See* Miscellaneous Utility Library
 - missing features of Stream 257
 - missing keys in Hashtables 132
 - mistakes 352
 - ML 231, 354
 - mocking 272
 - Mono 20, 25, 199
 - Monty Python 221
 - MouseClicked 139
 - MouseEventArgs 141
 - MouseEventHandler 55
 - MoveNext 162, 164, 166, 169, 172, 279

MSDN

- confusing join terminology 297
- EventArgs 141
- expression trees 239–240
- extension methods 10
- generic reflection 93–95
- LINQ standard query operators 281
- LINQ to XML 338, 342
- multi-core processors 357
- Multiply 239
- multi-threading 116, 178, 349
- mutability wrappers for value types 115
- myths 48, 51

N

-
- N+1 selects problem 325
 - name/value pairs 226
 - namespace aliases 193–194
 - namespace hierarchy 195
 - namespaces 261
 - for extension methods 262, 273
 - XML 338
 - naming 356
 - conventions for type parameters 71
 - NAND gates 127
 - native code 49, 87, 199, 239
 - native language of LINQ provider targets 244
 - natural language 271
 - natural result type 132
 - navigation via references 300
 - nested classes, restriction on extension methods 258
 - nested types 191
 - partial types 185
 - used for iterator implementation 164
 - .NET
 - influences 18
 - terminology 24
 - version numbers 27
 - .NET 1.0 20
 - .NET 1.1 20–21
 - .NET 2.0 21, 201, 232, 261
 - .NET 3.0 21
 - .NET 3.5 21, 208, 232, 261, 266, 277, 315, 344
 - extension methods 263, 289
 - .NET Passport 25
 - NetworkStream 257
 - new 224
 - anonymous types 57
 - with constructor type constraints 77
 - NHibernate 345
 - Criteria API 345
 - nodes 101

- NodeType 239
- non-abstract 76
- non-nullable constraint 323
- NOR gates 127
- Norås, Anders 272
- Novell 20
- null
 - argument when invoking static methods 95
 - as empty delegate invocation list 39
 - as failure value 131
 - behavior as an argument value 53
 - comparisons in generics 81–82
 - DBNull 114, 335
 - default value for reference types 82
 - extension methods 262
 - implicit typing 212
 - initial value in iterator blocks 172
 - nullable types 13, 58, 112–136
 - returned by Hashtable indexer 99
 - semantic meanings 114
 - target of extension methods 262–263, 273
 - type inference 223
 - null coalescing operator 128, 133
 - use with reference types 130
 - null literal 122
 - null value 335
 - See also* nullable types, null value
- Nullable class 119
 - Compare method 120
 - Equals method 120
 - GetUnderlyingType method 120
- nullable logic 127–128
- nullable types 58–59, 112–136, 258, 335
 - CLR support 118–119
 - conversions 124
 - conversions and operators 124–128
 - framework support 115–118
 - in generic comparisons 82
 - logical operators 127
 - not valid with value type constraints 76
 - novel uses 131–136
 - null value 120
 - operators 125
 - syntactic sugar in C# 2 120
 - underlying type 116, 120
 - Value property 127
- Nullable<T> 13, 116
 - equality 119
 - Equals behavior 119
 - explicit conversion to non-nullable type 117
 - GetHashCode behavior 116
 - HasValue property 116
 - implicit conversion from non-nullable type 117
 - struct 116
 - ToString behavior 117

Nullable<T> (*continued*)
 Value property 116
 wrapping and unwrapping 117
 nullity check 149
 NullReferenceException 82, 118–119, 262
 numbers 272
 numeric constraints 106
 numeric types 125

O

object creation 215
 object initializers 215–218, 283, 355
 object orientation, indirection 38
 Object Relational Mapping (ORM) 22, 351
 Entity Framework 348
 LINQ to SQL 314, 325
 NHibernate 345
 use of partial types 187
 Object Services 348
 object-orientation 20
 object-oriented data model 276, 300
 objects
 location in memory 51
 parameter passing 52
 ObjectSpaces 22
 obscure code 214
 Office 2007 21
 OfType. *See* Standard Query Operators, OfType
 open source 345
 open types 68
 closed at execution time 93
 operands, of null coalescing operator 128
 operator constraints 106
 operators 108, 191
 default. *See* default value expressions
 typeof. *See* typeof operator
 optimization 109, 242, 266
 of join ... select 300
 Option Explicit 47
 Option Strict 47, 355
 optional fields 130
 Oracle 346
 OrDefault 363
 OrderBy 10, 237
 orderby clauses. *See* query expressions, orderby
 clauses
 OrderBy. *See* Standard Query Operators, OrderBy
 OrderByDescending. *See* Standard Query
 Operators, OrderByDescending
 ordering 15, 267, 290
 grouping 308
 of object initialization 217
 ORM. *See* Object Relational Mapping (ORM)
 out 234, 258

outer sequences 297
 outer variables 151
 output parameters 81, 131
 Output window 198
 over-engineering 136
 overkill 12
 overloaded methods and generics 83
 overloading 290
 ambiguity 140, 252
 changes in C# 3 251–253
 extension methods 261
 GroupBy 310
 of Add method 220

P

P/Invoke 199
 pairs of values 83
 parallel computing 358
 Parallel Extensions 349, 357
 Parallel LINQ (PLINQ) 315, 349, 357
 ParallelEnumerable<T> 349
 parallelism 179, 357
 parallelization 313
 parameter 52
 covariance 230
 passing 51–52
 type contravariance XE 47
 type of object 64
 types in anonymous method declarations 146
 parameter wildcarding 149
 ParameterExpression 243
 parameterized typing. *See* generics
 ParameterizedThreadStart 150, 153
 parameterless constructors 76, 217
 entity classes 318
 parameters 151
 for delegates 35
 in expression trees 243
 positional 217
 params 20
 parentheses 234, 339
 Parse 342
 partial 185
 classes 317
 comparisons 134
 methods 188, 318
 modifier 189
 types 184–190, 258
 PartialComparer 135
 partition of the Common Language Infrastructure
 specification 25
 partitioning. *See* Standard Query Operators,
 partitioning
 pass by reference. *See* parameter passing

patterns 131, 136
 for comparison 135
 pausing, apparent effect in iterators 167
 PDC. *See* Professional Developers Conference
 peek 100
 performance 48, 111, 149, 190, 265–266, 291, 341
 improvements due to generics 65
 of value types 51
 permissions 203
 persistence layer 334
 personal preference for conditional operator 129
 personality 214
 philosophy of nullability 112
 photocopying, comparison with value types 49
 pinning 201
 pipelines 266
 placeholders, type parameters 68
 PLINQ. *See* Parallel LINQ (PLINQ)
 Point, as reference type or value type 50
 pointers 199
 pollution of classes with methods for
 delegates 145
 Pop 100
 portable 19
 positional parameters 217
 Postgres 346
 pragma directives 197, 199
 Predicate 97, 147, 236
 predicates 97, 291
 preprocessing query expressions 276, 284,
 286, 290
 PreserveOrdering 350
 primary keys 300
 primitive types 199
 private access modifier 193
 private variables, backing properties 209
 Process 222
 ProcessStartInfo 222
 producer/consumer pattern 101
 production assemblies 202
 productivity 214, 273
 Professional Developers Conference 19, 21
 project management 269
 project plan 175
 projection expressions 287, 307
 projection initializers 227, 269
 projection. *See* Standard Query Operators,
 projection
 projections 266, 283, 300, 303, 342, 356
 properties 57, 108, 212, 215, 340
 anonymous types 226
 automatic implementation 7
 automatically implemented. *See* automatically
 implemented properties
 comparison with events 40

 differing access for getter/setter 6
 forced equality of access in C# 1 6
 in value types 51
 separate access modifiers 192
 Properties window 197
 property descriptors 238
 property evaluations in expression trees 243
 provability 358
 pseudo-code 306
 pseudo-synchronous code 178
 public key 203
 token 203
 public variables 209
 Push 100
 Python 48

Q

qualified namespace aliases 194
 quantifiers. *See* Standard Query Operators,
 quantifiers
 query continuations 310
 query expression pattern 285
 query expressions 14, 275–314
 consistency across data sources 337
 continuations 310
 embedded within XEElement constructors 340
 from clauses 283, 303, 367
 functional style 354
 join ... into clauses 301
 let clauses 295, 321
 LINQ to SQL 325
 nested 341
 nesting 300
 orderby clauses 292
 Queryable extension methods 330
 range variables. *See* range variables
 readability 357
 select clauses 283, 287, 292, 300, 303
 translations 284
 transparent identifiers. *See* transparent
 identifiers
 type inference 246
 where clauses 290, 321, 332, 368
 query parameters 320
 query translation 318
 Queryable 263, 266, 326, 328, 330, 333, 362
 querying
 looping in C# 1 11
 separating concerns in C# 2 11
 QueryOptions 350
 QueryProvider 326
 Queue<T> 100
 quirks in iterator block implementation 172

R

-
- Rails 22
 - random access 102
 - random numbers 278
 - range 174
 - required functionality 174
 - range variables 287, 290, 295, 300, 304, 306, 310
 - explicitly typed 289
 - Range. *See* Standard Query Operators, Range
 - Range<T> 264, 268
 - Read 257
 - readability 12, 208
 - anonymous methods 138, 151, 159, 230
 - C# 3 initializers 217, 222–223
 - C++ templates 108
 - extension methods 260, 263, 266, 270, 272
 - generic method type inference 80
 - generics 111
 - implicit typing 213
 - lambda expressions 10, 232, 234
 - namespace aliases 193
 - null coalescing operator 130–131, 133
 - of implementation 356
 - of nullable type syntax 120
 - of results 356
 - query expressions 291, 308, 312
 - ReadFully 257
 - ReadOnly 98
 - read-only
 - automatically implemented properties 210
 - properties 215
 - recursion, XElement constructor 339
 - redundancy of specifying type arguments 79
 - ref 234, 258
 - refactoring 21, 85, 133, 187, 316, 356
 - reference 113
 - obtained via boxing 53
 - passed a parameter 52
 - reference parameters 210
 - reference types 48, 75
 - use with null coalescing operator 130
 - ReferenceCompare 135
 - reflection 77, 92, 238, 244, 340, 355
 - Reflector 123, 146, 172, 292
 - register optimizations 88
 - relational data model 300
 - relational model 276
 - relational operators 125
 - on nullable types 125
 - reliability 21
 - remote execution 245
 - Remove 38, 138
 - RemoveAll 97
 - RemoveAt 101
 - Repeat 98
 - Repeat. *See* Standard Query Operators, Repeat
 - Reset 172
 - resource requirements 175
 - responsibilities in design 187
 - restoring warnings 198
 - restraint 208
 - return keyword, anonymous methods 148
 - return statements 169, 233, 253
 - prohibited within iterator blocks 166
 - return type
 - covariance 46
 - inference 288
 - of object 64
 - requirement for delegate matching 35
 - return types 108
 - covariance 141
 - covariance for delegates 143
 - lambda expressions 233
 - methods differing only by 90
 - of partial methods 189
 - type inference 247
 - return values 131
 - Reverse. *See* Standard Query Operators, Reverse
 - reversing 265
 - of ranges 175
 - RIA. *See* Rich Internet Applications (RIA)
 - Rich Internet Applications (RIA) 22, 24, 358
 - richness of libraries increasing with extension
 - methods 271
 - right associativity of null coalescing operator 130
 - Robotics Studio 178
 - robustness 21, 194
 - Ruby 22, 48
 - runtime 25
 - cast checking 51
-
- S**
-
- safety 111
 - sample data 283
 - scalability 181
 - schema 316
 - LDAP 347
 - Scheme 151
 - scientific computing 106
 - scope 151, 155
 - of finally blocks in iterators 171
 - range variables 311
 - scripting 24
 - SDK. *See* Software Development Kit (SDK)
 - sealed 255
 - implicit for static classes 191
 - sealed types 190
 - security 203

- security permissions 319
- seed aggregation 359
- select clauses. *See* query expressions, select clauses
- Select, method of DataTable 334
- Select. *See* Standard Query Operators, Select
- SelectMany. *See* Standard Query Operators, SelectMany
- semi-colons 234
- sender events 237
- sequence diagram 279
- SequenceEqual. *See* Standard Query Operators, SequenceEqual
- sequences 278, 290
 - embedded 301
 - flattening 306
 - generated by multiple from clauses 306
 - infinite 278, 294
 - inner and outer 297
 - untyped 289, 361
- serialization 226
- Server Explorer 316, 335
- Service Oriented Architecture (SOA) 358
- servlet 18
- sessions, NHibernate 346
- set-based operations. *See* Standard Query Operators, set-based operations
- SetRange 98
- setter 192
- SharePoint 344
- sharing variables between anonymous methods 158
- short-circuiting operators 127, 133
- shorthand for nullable types 59
- signatures
 - complicated generics 250
 - of event handlers 142
 - of lifted operators 125
 - of methods 35
 - overloading in C# 3 251
 - real vs effective 288
- signed assemblies 203
- sign-off 175
- Silverlight 22
- simplicity 151, 159, 175, 214
- single inheritance of implementation 256
- single responsibility 163, 187
- Single. *See* Standard Query Operators, Single
- SingleOrDefault. *See* Standard Query Operators, SingleOrDefault
- singleton pattern 190
- SkeetySoft 282, 289, 307
- Skip. *See* Standard Query Operators, Skip
- SkipWhile. *See* Standard Query Operators, SkipWhile
- sloppiness 353
- snippets 28–30, 264
 - expansions 28
 - meaning of ellipsis 29
- Snippy 30, 144, 187, 241, 263
- SOA. *See* Service Oriented Architecture (SOA)
- software development 352
- Software Development Kit (SDK) 123
- software engineering 274
- Sort 105, 148, 236, 369
- SortedDictionary<TKey,TValue> 101
- SortedList 101
- SortedList<TKey,TValue> 101
- sorting 133, 265, 267, 356
 - files by name and size 148
 - in C# 1 8
 - See also* Standard Query Operators, sorting
- source assembly 201
- source code 25
 - available to download 30
- source files 198
- Spec# 21
- special types, restrictions in derivation type
 - constraints 78
- specialization 255, 356
 - of templates 109
- specification 25
 - Common Language Infrastructure 25
 - See also* language specification
- speculation 352
- SQL 15, 22, 275–276, 351
 - behavior of NULL values 128
 - joins 294, 300–301, 304, 322
 - LINQ to SQL 244, 314, 325
 - logging 318
 - ordering of clauses 286
- SQL Server 346, 348
- SQL Server 2005 21, 315
 - Express edition 316
- SQL Server 2008 348
- SQL Server Management Studio Express 322
- square brackets 94
- square root 267
- stable sort 369
- stack 50, 52, 54, 155–156, 201
- stack frame 152, 155
- Stack<T> 100
- Standard Query Operators 274, 281, 327, 343, 359–370
 - Aggregate 359
 - aggregation 332, 359
 - All 367
 - Any 367
 - AsEnumerable 360
 - AsQueryable 360
 - Average 333, 359

Standard Query Operators (*continued*)

- Cast 289, 326, 361
- Concat 360
- concatenation 360
- Contains 367
- conversion 360
- Count 301, 333, 359
- DefaultIfEmpty 323
- Distinct 368
- element operations 363
- ElementAt 363
- ElementAtOrElse 363
- Empty 364
- equality operations 363
- Except 368
- filtering 368
- First 363
- FirstOrElse 363
- generation 364
- GroupBy 269, 308, 365
- grouping 365
- GroupJoin 303, 365
- Intersect 368
- Join 365
- joins 365
- Last 363
- LastOrElse 363
- LongCount 359
- Max 359
- Min 359
- OfType 289, 326, 361, 368
- OrderBy 267, 271, 294, 369
- OrderByDescending 267, 294, 369
- partitioning 366
- quantifiers 367
- Range 264, 364
- Repeat 364
- Reverse 264, 281, 369
- Select 266, 277, 281, 285, 292, 296, 367
- SelectMany 306, 367
- SequenceEqual 363
- set-based operations 368
- Single 320, 363
- SingleOrDefault 363
- Skip 366
- SkipWhile 366
- sorting 369
- Sum 269, 332, 359
- Take 366
- TakeWhile 366
- ThenBy 267, 271, 294, 369
- ThenByDescending 267, 294, 369
- ToArray 360
- ToDictionary 361
- ToList 360
- ToLookup 361, 365
- Union 368
- Where 265, 285, 291, 368
- standardization 19
- StartsWith 243
- state
 - lack of in utility classes 190
 - requirement for iterators 163
- state machine 181
 - for iterators 166
- static classes 190–197, 256, 258, 356
 - System.Nullable 119
- static constructors 87, 221
- static generic methods, workaround for lack of
 - generic constructors 108
- static languages 354
- static members
 - of generic types 85
 - thread safety 209
- static methods 190, 255, 258
 - as action of delegate 36
- static modifier for static classes 191
- static properties 209
- static typing 42–43, 57, 65, 141, 211, 354
 - compiler enforcement 43
 - of Expression 240
- StaticRandom 349
- storage 278
- Stream 51, 142, 257, 278
- streaming 265, 279, 289–290, 306, 308
- streamlining 353
- StreamUtil 257
- string
 - as a reference type 50
 - as example of immutability 39
- StringBuilder 52
- StringCollection 46
- StringComparer 364
- StringDictionary 99
- StringProcessor 34
- strong typing 42, 45, 56
 - of implementation 74
- strongly typed interface 85
- Stroustrup, Bjarne 109
- struct. *See* value types
- structs as lightweight classes 51
- structure 222, 339
- structures as generic types 67
- SubmitChanges 319
- subtraction of nullable values 123
- Sum. *See* Standard Query Operators, Sum
- Sun 18, 20, 24
- symmetry 135
- Synchronized 98
- SynchronizedCollection<T> 99–100

SynchronizedKeyedCollection 71, 100
 synchronous code 179
 syntactic shortcut for anonymous methods 149
 syntactic sugar 13, 162, 181, 216, 276
 for nullable types 120
 syntax 25
 System namespace 232
 System.Linq namespace 263
 System.Linq.Expressions namespace 239
 System.Nullable class. *See* Nullable class
 System.Nullable<T> struct. *See* Nullable<T> struct
 System.Runtime.CompilerServices namespace 261
 System.Web.UI.WebControls 193
 System.Windows.Forms 193
 System.Xml.Linq 338

T

Table 319, 326
 Take. *See* Standard Query Operators, Take
 TakeWhile. *See* Standard Query Operators, Take-While
 Tatham, Simon 182
 TDD. *See* Test-Driven Development (TDD)
 technology
 history 18
 milestones 22
 pace of change 3
 tedious coding 229
 template arguments 109
 template metaprogramming 109
 templates 20, 83, 102, 108
 terminology 253, 258, 297
 automatically implemented properties 209
 ternary operator. *See* conditional operator
 test-bed for captured variables behavior 160
 Test-Driven Development (TDD) 174
 text editor 222
 TextReader 278
 ThenBy. *See* Standard Query Operators, ThenBy
 ThenByDescending. *See* Standard Query Operators, ThenByDescending
 thinking in C# 18
 third party libraries 273
 this 151, 155, 163, 221, 258, 329
 Thread constructor 140
 thread safe 39
 thread safety 39, 116, 209, 226
 uses for queues 100
 threadpool 153
 threads 153, 178
 ThreadStart 34, 140, 143, 150, 152
 throwaway code 209
 TIME_ZONE_INFORMATION 199
 TimeSpan 116, 125
 ToArray. *See* Standard Query Operators, ToArray
 ToDictionary. *See* Standard Query Operators, ToDictionary
 ToList 236
 See also Standard Query Operators, ToList
 ToLookup. *See* Standard Query Operators, ToLookup
 tooltips 211, 225, 260
 ToString 53, 226, 238, 240, 268, 284, 318, 329
 traditional managed developers 24
 transaction management 318
 transactions, NHibernate 346
 transformations 161, 279
 translations 284
 transparent identifiers 294, 300, 306
 erasure 297
 trial and error 66
 TrimExcess 98
 TrimToSize 98
 tri-state logic 127
 trivial key selector 298
 trivial properties 209, 228–229
 Trojan horse 24
 true operator 125
 TrueForAll 96
 truth tables 127
 TryGetValue 99
 TryParse 81, 132
 TryXXX pattern 81, 131
 T-SQL 316
 two-phase type inference 248
 type arguments 68, 74, 92, 246
 inference 79
 of anonymous types 225
 type constraints 74–75, 185
 combining 78
 lack of operator/numeric 106
 restricted combinations 78
 type conversions and type safety 44
 type declarations 185
 type erasure 110
 type hierarchy expression classes 240
 type inference 43, 79, 288, 290, 355
 anonymous types 225
 changes in C# 3 245–253
 in C# 2 74
 local variables 210
 return types of anonymous functions 247
 two-phase in C# 3 248
 type initialization 85
 type parameter constraints 77
 type parameters 67, 92, 109, 211
 and typeof 93
 constraints. *See* type constraints

type parameters (*continued*)
 inference 80
 unconstrained 75
 Type property of Expression 239
 type safety 42, 44, 103, 111
 type system
 characteristics of C# 42, 48
 in C# 1 33
 richness 45
 summary of C# 1 48
 type systems 42, 59
 type variables, fixed and unfixed 248
 typed datasets 335–338
 typedef 106
 TypedTableBase 337
 typeless expressions 223
 typeof 244
 with nullable types 121
 typeof operator 69, 81, 92
 type-specific optimizations 109
 typing, IComparer<T> 9

U

unary logical negation operator
 for bool? 127
 unary operators 125
 unbound generic types 68, 92
 unboxing 53–54, 89
 of Nullable. *See* Nullable, boxing and unboxing
 uncertainty 121
 unconstrained type parameters 75
 undefined behavior of IEnumerator.Current 173
 unfixed type variables 248
 unified data access 276
 unintended comparisons with nullable values 126
 Union. *See* Standard Query Operators, Union
 unit testing 202, 353
 unit tests 131, 222, 272
 Unix 19
 unknown values 13
 UnmanagedType 201
 unsafe code 42, 199
 unsafe conversions 44
 unsigned assemblies 203
 untyped datasets 334–335
 unwrapping 123
 UPDATE 319
 Update 319
 URLs
 comparison with references 49, 54, 113
 fetching in new threads 153
 user interface 269
 user-defined operators 125
 using directive 135, 193, 261

using statement 25, 28, 181, 193
 implicit typing 213
 UTF-8 44
 utility classes 190, 256, 266

V

validation 188, 192, 209, 273
 Value
 property of DictionaryEntry 67
 property of Nullable<T>. *See* Nullable<T>, Value
 property
 value type constraints 76
 on Nullable 116
 value types 48, 178, 255
 automatic properties 210
 generic type parameters 76
 handled efficiently with generics 65
 non-nullability 58, 113
 Values 101
 var 43, 211
 See also implicit typing
 varargs 20
 variable
 impossibility of hiding in C# 1 6
 untyped 43
 variables
 definite assignment 65
 implicitly typed 57
 instance 50
 local 50
 location in memory 50–52
 range variables 287
 static 51
 Variant 211
 VB Classic 20
 VB.NET 20, 24, 201
 VB6 20, 24, 211
 VB8 21
 VB9 277, 281
 version number 26
 versioning 110, 261, 273
 virtual methods 184, 188
 visibility 356
 visible constructors 191
 Vista 21, 200
 Visual Basic 47, 355
 version numbers 27
 Visual C++ team 109
 Visual Studio .NET 2002 20
 Visual Studio .NET 2003 20
 Visual Studio 2005 21, 187, 197
 Visual Studio 2008 22, 263
 DataSet designer 335
 displaying extension methods 260

Visual Studio 2008 (*continued*)
 displaying inferred types 211, 225
 Intellisense in LINQ 325
 LINQ to SQL designer 316
 reference properties window 197
 update for Entity Framework 348
 Visual Studio, version numbers 27
 void 147, 189

W

warning numbers 198
 warning pragmas 197
 warnings 144, 197
 weak typing 42, 211
 of interface 74
 web page, comparison with reference types 49
 web service proxies 187
 web services 179, 269, 344, 346
 web site of book 30
 WebRequest 258
 WebResponse 258
 where 15
 type constraints 75
 Where. *See* Standard Query Operators, Where
 whitespace 147, 203, 218–219, 267
 wiki 49
 wildcards 110
 will 34, 38
 Windows 19, 26, 199, 347
 Windows Forms 26, 184
 Windows Live ID 24
 Windows Presentation Foundation
 used to write Snippy 30
 Windows Presentation Foundation (WPF) 21, 187
 Windows Server 2008 200
 workflow, of iterators 167
 WPF. *See* Windows Presentation Foundation
 (WPF)

wrapper
 for faking nullity 13
 for value types 115
 write once, run anywhere 19
 WriteLine 67

X

XAML. *See* Extensible Application Markup Language (XAML)
 XAttribute 338, 340, 342
 XComment 338
 XDocument 338–339, 343
 XElement 338–339, 342
 axis methods 342
 XML 16, 222, 277, 338
 declaration 339
 documentation 209
 mapping files for ADO.NET Entity
 Framework 348
 metadata for LINQ to SQL 316
 used to describe standard libraries 26
 XName 338
 XPath 343

Y

yield statements 165
 finally blocks 170–172
 restrictions 166
 yield break 169–170
 yield return 165–169
 yield type 166

Z

zero
 representation of null 113
 result of comparison 134