

# Index

## Symbols

& (ampersand) operator, 65  
&& (double ampersand) operator, 65–66  
\* (asterisk)  
  as arithmetic operator, 57  
  in Forms Designer window, 24  
\  
  (backslash), special characters and, 95  
{ (braces)  
  class name and, 102–103  
  using for clarity, 72  
[] (brackets)  
  array and, 112  
  as index operator, CD76  
} (close brace) expected error message, 377  
= (equals) sign  
  as assignment operator, 40, 60–61  
  reference types and, 107–108  
! (exclamation point) operator, 65  
^ (exclusive or—xor) operator, 65  
// (forward slashes), 34  
- (minus) sign and code region, 31  
% (modulo) operator, 58, CD60  
| (pipe) operator, 65  
|| (double pipe) operator, 66  
+ (plus) operator and strings, 52  
+ (plus) sign and code region, 31  
<T> and generic collections and, 338

/// (three-slash) comment, 181, CD117  
~ (tilde), 271

• A •

absolute value function, 64  
abstract class  
  declaring, 319  
  overview of, 293–294  
  using, 294–296  
abstracting concepts, 135  
AbstractInheritance program, 294–295  
AbstractInterface program, 316–319  
abstraction. *See also* abstract class  
  class factoring, 288–293  
  overview of, 213–215, 288  
AcceptButton property, 24  
access control  
  accessor methods, 226–227, 231  
  containment and, 258–259  
  DoubleBankAccount program example, 227–230  
  importance of, 225–226  
  overview of, 218–219, 221–224  
  security, levels of, 224–225  
accessing  
  collection, CD72–CD74  
  current object, 169–176  
  member of object, 104–106  
  project properties, 160–161  
  static member of class, 110  
alignment guides, 22  
AlignOutput program, 201–203  
Alt+Tab (switch program), CD129

ampersand, double (&&) operator, 65–66  
ampersand (&) operator, 65  
application. *See also* console application; *specific programs*  
  action, adding, 25–27  
  breaking, CD132–CD135  
  building and running, 18–20  
  console, creating, 29–31  
  converting class into, CD114–CD115  
  creating, 15  
  description of, 12  
  developing, CD142  
  dividing into multiple assemblies, CD29–CD30  
  dividing into multiple source files, CD28–CD29  
  executable, 12, 373  
  executing, 19, 32, 35  
  Forms Designer and, 20–24  
  freeware, 388  
  rebuilding and running, 24–25, 373–374  
  running on different machines, CD180  
  shareware, 388  
  source files for example, 387–388  
  template, creating, 15–18  
  testing, 27–28  
Application Wizard, 16, 17, CD32  
approximation error, CD3  
argument  
  auto-complete feature and, 178–179  
  implementing default, 140–142  
  matching definitions with usage, 138–139

- argument (*continued*)
    - multiple, passing to function, 136–138
    - as part of name of function, 274–275
    - passing from DOS prompt, 155–157
    - passing from Visual Studio 2005, 159–162
    - passing from window, 157–159
    - passing to default base class constructor, 266–269
    - passing to function, 136
    - value-type, passing by reference, 143–147
    - value-type, passing by value, 142–143
  - arithmetic operators
    - assignment, 60–61
    - increment, 61–62
    - operating orders, 58–59
    - overview of, 57
    - simple, 57–58
  - array
    - argument for, 112
    - description of, 101, 111–112
    - disadvantages of, 334
    - fixed-value, 112–114
    - iterating through, 192
    - Length property, 117
    - linked list compared to, CD62
    - naming, 120
    - objects and, 118–120
    - sorting elements within, 122–126
    - variable-length, 114–118
  - ArrayList class, 334, 335–336
  - as operator, 264–265, 337
  - assembly, CD29–CD30
  - assigning
    - expression type, 68–69
    - multiple catch blocks, CD15–CD17
  - assignment of reference, 126
  - assignment operator (=)
    - declaring variable and, 40
    - math functions and, 60–61
    - reference types and, 107–108
  - asterisk (\*)
    - as arithmetic operator, 57
    - in Forms Designer window, 24
  - author, Web site of, 8, 390
  - “auto list members” Help, CD124–CD125
  - auto-complete feature
    - displaying, 26, 27
    - documentation comments and, 180–184
    - overview of, 176–177
  - System Library and, 177–179
  - for user-created functions and methods, 179–180
  - auto-indenting, 75
  - AverageAndDisplay()
    - function, 137–138
  - AverageAndDisplay Overloaded program, 139–140
  - AverageStudentGPA program, 119–120
  - AverageWithCompiler Error program, 138
  - avoiding
    - boxing and unboxing, 331
    - confusion in code, 59, 60
    - duplication among constructors, 245–248
    - else statement, 76–77
    - goto statement, 98
    - redundancy, 292
    - tight coupling, 359
- B •**
- backslash (\) and special characters, 95
  - BankAccount class, inheriting from, 254–257
  - BankAccount program, 222–224
  - BankAccount ConstructorsAnd Function program, 245–247
  - BankAccount ConstructorsAndThis program, 247–248
  - BankAccountWith Multiple Constructors program, 243–245
  - base class method, overloading, 275–280, 374–375
  - base interface, 316
  - base keyword, 268–269, 280–281
  - binary operator, 58
  - bitwise operator, 65
  - blank line, generating, 106
  - bonus chapters, 388
  - bool variable, 49, 64–66
  - boxing
    - nongeneric collections and, 336, 337
    - value-type variable and, 149, 330–331
  - braces ({} )
    - class name and, 102–103
    - using for clarity, 72
  - brackets ([])
    - array and, 112
    - as index operator, CD76
  - break command
    - looping statements and, 84–86
    - nested loop and, 93
    - while loop and, 86–89
  - breaking
    - encapsulation, 279
    - string, 203–205
  - breakpoint, setting, 238, 239, CD132–CD135
  - browsing online help, 177
  - bubble sort, 123–126
  - buffer overwrite, CD72
  - BuildASentence program, 190–192
  - building Windows Forms program, 18–20
  - Button control, 22
- C •**
- C language, 61
  - C++ programming language
    - constructors and, 382–383
    - global data or functions and, 380

- interface and, 383–384
  - memory allocation and, 380
  - multiple inheriting and, 383
  - overview of, 379–380
  - pointer variables and, 381
  - template feature, 381–382
  - type checking and, 382
  - Unified Type System, 384
  - variable types and, 383
- C# version 2.0**
- features of, 2
  - overview of, 12–13
  - resources on, 7–8, 177, 382
- CalculateInterest**
- program, 73–74
- CalculateInterestTable**
- program, 81–82, 129
- CalculateInterestTable**
- MoreForgiving**
  - program, 86–87
- CalculateInterestTable**
- WithFunctions**
  - program, 130–132
- CalculateInterestWith**
- EmbeddedTest**
  - program, 77–78
- calculating**
- interest, 73
  - type of operation, 67–68
- call by reference feature,**
- 147, 148–149
- Call Stack window,**
- CD135–CD136
- calling initialization**
- function, 233
- CAN\_BE\_USED\_AS**
- relationship
  - overview of, 303–305
  - program example of, 307–316
- Cannot implicitly convert**
- type ‘x’ into ‘y’ error
  - message, 369–371
- cast**
- as operator and, 264–265
  - description of, 56
  - invalid, 262–263
  - protecting with `is`
  - operator, 263
- cast operator,** 68, 69
- catch blocks, assigning**
- multiple, CD15–CD17
- CD-ROM**
- contents of, 385, 387–389
  - description of, 6
  - installing items to hard
  - drive from, 386–387
  - system requirements for, 385–386
  - temperature conversion
  - program, 44, 46
  - troubleshooting, 389–390
- chaining assignments,** 60
- changing type of variable,**
- 55–56
- chapters, bonus,** 388
- char variable,** 50–51, 53
- character**
- adding to or removing from
  - end of string, 201–203
  - escaping, 95
  - newline, 106
  - parsing out of string, 194–196
  - replacing one with another, 203–205
- character types,** 50–52
- class. *See also* Priority**
- Queue** class
  - abstract, 293–296, 319
  - access control and, 219, 221–224
  - adding, CD110–CD111
  - ArrayList**, 334, 335–336
  - BankAccount**, inheriting
  - from, 254–257
  - collection, writing own, CD61–CD71
  - concrete, 295
  - Console**, 160
  - const data member, 111
  - containing another class, 108–109
  - converting into program, CD114–CD115
  - CustomException**, CD14
  - declaring abstract, 319
  - defining, 102–103
  - description of, 101, 102, 233
  - Dictionary**, 335
  - Exception**, creating own, CD13–CD15
  - Exception**, overriding, CD22–CD26
- factoring,** 288–293, CD118
- FileStream**, CD43–CD44
- functions and,** 163
- generic, methods in, 353–355
- hierarchy of inherited,** 266
- hierarchy, restarting,**
- 296–299
- implementation of interface**
- and, 305
- inheritance and,** 252–253
- instantiating locally,** 249
- internal changes to,** 230
- internal member of,** 225
- internal protected**
- member of, 225
- LinkedList**, 334
- methods and,** 163
- name of,** 102–103
- namespace and,**
- CD33–CD35
- nongeneric collection,**
- 334–336
- non-static member of,** 128
- Object**, 327, 328, 330
- object,** 264
- of object, changing,** 261–262
- object of,** 103–104
- as part of name of function,**
- 274–275
- partial,** CD177–CD178
- protected member of,** 225
- public member of,**
- 221–224, 225
- Queue**, 334
- as responsible for itself,** 279
- sealing,** 300, 375
- Simple Factory**, 358–359
- Stack**, 335
- static member of,** 110
- StreamReader**,
- CD50–CD54
- StreamWriter**,
- CD45–CD50
- String**, 187
- structure compared to,** 327
- wrapper,** 342
- class constructor,** 235–236
- class function,** 128
- class inheritance,** 219
- class library,** CD27, CD29, CD39–CD43

- class method, 168
- class property, 110, 231–232, CD91
- Class View, CD110
- classification, 215–217
- ClassLibrary program, CD40–CD41
- ‘className’ does not implement interface member ‘methodName’ error message, 376
- ‘className.memberName’ is inaccessible due to its protection level error message, 371–372
- Click event, setting, 25–26
- close brace (}) expected error message, 377
- closed window, CD100
- closing program, 20
- CLR (Common Language Runtime), 3
- code
  - navigating, CD110
  - testing on different machines, CD180
  - for Windows forms, writing, CD175–CD179
  - writing, CD115–CD119
- code region, 31
- Code Snippets feature, 34–35
- Code view, 25
- collection. *See also* generic collections
  - as alternative to array, 117
  - description of, CD86–CD87
  - directory of files, iterating through, CD55–CD61
  - index feature for, CD76–CD80
  - iterating, CD72–CD75
  - nongeneric, 334–336
- collection class, writing own, CD61–CD71
- commands
  - break, 84–89, 93
  - Console.Write(), 330
  - continue, 84–86
  - Readline(), 74
  - ToDecimal(), 74
  - WriteLine(), 74, 160
- comments
  - documentation comments, 180–184
  - reading, 129
  - using, 34, CD117
- Common Language Runtime (CLR), 3
- Community menu, CD124
- Compare() method, 189–193
- comparing floating point numbers, 47, 63–64
- compiler error, 336–337
- composing, 259
- compound assignment operators, 61
- compound logical operators, 64–66
- computational speed with decimal variables, 49
- Concat() method, 204–205
- concept
  - class and, 102
  - function and, 135
- concrete class, 295
- console application
  - comments in, 34
  - core of, 34–35
  - dropping file onto, 157, 158–159
  - editing template file, 32–33
  - executing by double-clicking, 157–158
  - framework for, 33
  - I/O and, CD43
  - overview of, 29
  - template, creating, 29–31
  - testing, 31
- Console class, 160
- Console.Write()
  - command, 330
- Console.WriteLine statement, 34–35
- const data member, 111
- constant
  - declaring, 79, 111
  - numeric, declaring, 54–55
  - string constant, 189
  - writing, CD6
- constraint
  - generic method and, 356
  - PriorityQueue() class and, 351–352
  - of type parameter, 362
- constructor
  - avoiding duplication among, 245–248
  - C++ and, 382–383
  - C#-provided, 233–234
  - default, 235–237
  - default base class, invoking, 265–266
  - default, initializing object directly and, 241–242
  - default, passing arguments to, 266–269
  - default, program example, 236–237
  - DemonstrateDefault Constructor program, 236–237
  - description of, 233, 234
  - executing from debugger, 238–241
  - initializing object directly and, 241–242
  - overloading, 243–245
  - parameters for, 362–364
- ConstructorSavings Account program, 269–271
- containers
  - iterating through, 120–121
  - linked list, CD61–CD71
- containment, 258–259
- continue command, 84–86
- control window, 21
- controlling output manually
  - Format() method, 206–210
  - overview of, 200–201
  - Trim() and Pad() methods, 201–203
- controlling program flow
  - for loop, 90–92
  - foreach statement, 120–121
  - goto statement, 98
  - if statement, 72–79
  - nested loops, 92–95
  - switch control, 96–98
  - while loop, 80–90
- controls
  - events and, 25–27
  - properties of, 23–24
  - putting on form with code only, CD178–CD179
  - working with, 21–22

- Convert class, 196, 198
  - converting variable type, 55–56
  - counting
    - decimal variable and, 49
    - floating point variable and, 46
  - counting variable
    - in for loop, 92
    - incrementing in while loop, 83–84
  - CPU processor fault, 153
  - Ctrl+C (interrupt running program), CD56
  - Ctrl+F5 (Start Without Debugging), 159
  - Ctrl+K then Ctrl+X (view menu of snippets), 34
  - Ctrl-Space (display auto-complete), 26
  - current object, accessing
    - overview of, 169–171
    - this keyword and, 171–176
  - CustomException class, CD14
  - CustomException program, CD23–CD26
- D •**
- data member, initializing, 172
  - data, storing, 101
  - data structure, CD61
  - Debug→Delete All
    - Breakpoints, CD134
  - debugger
    - executing constructor from, 238–241
    - SharpDevelop and, CD146–CD149
  - debugging tools
    - breakpoint, setting, CD132–CD135
    - overview of, CD126
    - single stepping, CD128–CD131
    - stack trace, CD135–CD136
  - Debug→Start Debugging, 159
  - Debug→Start Without Debugging, 19, 159
  - Debug→Step Into, 238
  - Debug→Step Over, CD128
  - Debug→Windows→
    - Breakpoints, CD134
  - Debug→Windows→Call Stack, CD135
  - decimal variable, 48–49
  - DecimalBankAccount program, 228–230
  - Decimal.Round() method, 228–230
  - decimal.Round() statement, 83
  - decision-making. *See* controlling program flow
  - declaration, 41
  - declared type, 283–284
  - declaring
    - bool variable, 49
    - class abstract, 319
    - constant, 79, 111
    - decimal variable, 48
    - floating point variable, 45–46
    - namespace, CD31–CD32
    - numeric constant, 54–55
    - variable, 40, 42, 319–320, 368, 369
  - decrement operator, 62
  - default constructor
    - initializing object directly and, 241–242
    - invoking, 265–266
    - overview of, 235–236
    - passing arguments to, 266–269
    - program example, 236–237
  - default location for folder, changing, 18
  - defining
    - class, 102–103
    - class properties, 231–232
    - function with no value, 152–153
    - method, 167–169
    - static member function, 165–167
    - variable type, 383
  - delegating, 259
  - DemonstrateDefaultConstructor program, 236–237
  - demotion, 67–68
  - dependency, 359
  - Dequeue() method, 342, 350
  - destructor, 271–272
  - Dictionary class, 335
  - directory
    - of files, iterating through, CD55–CD61
    - naming, 156
  - discriminating between objects, 106–107
  - DisplayArguments program, 153–154
  - displaying
    - auto-complete, 26
    - line numbers, CD135–CD136, CD137
    - project, CD108–CD110
    - zero, trailing, 83
  - DisplayRatio() function, 152–153
  - DisplayRounded
    - Decimal() function, 141
  - DisplayWithNestedLoops program, 93–94
  - distributed development, 13
  - dividing program
    - into multiple assemblies, CD29–CD30
    - into multiple source files, CD28–CD29
  - .DLL file extension, CD29
  - docked window, CD102
  - Document Explorer, CD119
  - document window, 21
  - documentation comments, 180–184
  - DOS command line
    - executing program from, 32
    - passing argument from, 155–157
  - DOS window, opening, 33
  - dot operator, 107–108
  - DotGNU Portable .NET implementation, 1
  - double ampersand (&&) operator, 65–66
  - double pipe (|) operator, 66
  - double variable, 45
  - DoubleBankAccount program, 227–230
  - do...while loop, 84
  - down conversion, 68

driver for project, creating, CD41–CD43  
 dropping file onto console program, 157, 158–159  
 duplication among  
   constructors, avoiding, 245–248  
 Dynamic Help window, CD124

## • E •

early binding, 283  
 elegance in software, 261  
 else statement, 75–77  
 embedded statement, 77  
 empty string, 52, 150  
 enabling Treat Warnings as Errors, 279  
 encapsulation, 279, 342. *See also* access control  
 Enqueue() method, 342, 349–350  
 enumerated type, 346  
 enumerator for linked list, implementing, CD69–CD71  
 equals (=) sign  
   as assignment operator, 40  
   reference types and, 107–108  
 error. *See also* run-time error  
   approximation, CD3  
   compiler, 336–337  
   rethrowing, CD20–CD21  
   returning, CD1–3, CD7–8  
 error message  
   Cannot implicitly convert type 'x' into 'y,' 369–371  
   'className' does not implement interface member 'methodName,' 376  
   'className.memberName' is inaccessible due to its protection level, 371–372  
   } (close brace) expected, 377  
   fixed-value array and, 113  
   'methodName': not all code paths return a value, 376–377

The name 'memberName' does not exist in the class or namespace 'className,' 368–369  
 overview of, 367  
 'subclassName': cannot inherit from sealed class 'baseclassName,' 375  
 'subclassName.methodName' hides inherited member  
 'baseclassName.methodName.' Use the new keyword if hiding was intended, 374–375  
 for two methods with identical name, 279  
 Unable to copy the file 'programName.exe' to 'programName.exe.' The process cannot..., 373–374  
 Use of unassigned local variable 'n,' 372–373  
 escaping character, 95  
 event, 25–27  
 exception  
   description of, 196  
   responding to, CD21–CD22  
   throwing, CD17–CD19  
 Exception class  
   creating own, CD13–CD15  
   overriding, CD22–CD26  
 exception mechanism  
   example of, CD10–CD13  
   overview of, CD9–CD10  
 exclamation point (!)  
   operator, 65  
 exclusive or—xor (^)  
   operator, 65  
 executable program, 12, 373, CD29  
 executable statement, CD128  
 executing  
   console program by double-clicking, 157–158  
   constructor from debugger, 238–241  
   program, 19, 32, 35  
   Visual Studio, 16  
 explicit type conversion, 68

expression type  
   assigning, 68–69  
   calculating, 67–68  
   matching, 66  
 eXtensible Markup Language (XML), 180–185  
 external method, 224

## • F •

FactorialErrorReturn program, CD4–CD7  
 FactorialException program, CD11–CD13  
 FactorialWithError program, CD2–CD3  
 factoring, 288–293, CD118  
 factory  
   generic, building, 359–364  
   overview of, 348, 358–359  
 F11 (Step Into), 238, CD130  
 F5 (Start Debugging), 159  
 file  
   include, 382  
   naming, 156  
   project, CD29, CD107  
   searching for, 155–156  
   source, 12, 387–388, CD28–CD29, CD30–CD38  
 file access, CD27  
 file extension  
   .cs, 12, 16  
   .DLL, CD29  
   .exe, 12, 383  
 File→New→Project, 16  
 FileRead program, CD50–CD54  
 FileStream class, CD43–CD44  
 FileWrite program, CD45–CD50  
 FixedArrayAverage program, 113, 114  
 fixed-length variable type, 52  
 fixed-value array, 112–114, 117–118  
 float variable, 45, 47  
 floating point numbers, comparing, 47, 63–64

- floating point variable
  - converting temperature using, 46
  - declaring, 45–46
  - description of, 44–45
  - limitations of, 46–49
- floating window, CD100, CD101
- flow control
  - for loop, 90–92
  - foreach statement, 120–121
  - goto statement, 98
  - if statement, 72–79
  - nested loops, 92–95
  - switch control, 96–98
  - types of, 72
  - while loop, 80–90
- F1 (Help), CD119–CD120
- for loop, 90–92
- foreach statement, 120–121, CD72–CD75
- form
  - control, adding to, 22
  - description of, 19–20
  - properties, changing, 23–24
  - writing own code for, CD175–CD179
- format controls, 207–208
- Format() method, 206–210
- format string, 160, 206
- formatting output, 206–210.
  - See also controlling output manually
- Forms Designer, 19–22, 28
- forward slashes (//), 34
- fractions, 43–44
- freeware program, 388
- Frequently Asked Questions sites, 8
- F10 (Step Over), CD128
- fully qualified name, CD34–CD38
- function call, 74
- functional programming, 214–215, 216–217
- functions. *See also* Main() function
  - absolute value, 64
  - advantages of, 133–134
  - auto-complete feature and, 179–180
  - AverageAndDisplay(), 137–138
  - built-in, 177–179
  - C++ programming language and, 380
  - calling initialization, 233
  - calling themselves, 280–281
  - class and, 163
  - concept and, 135
  - defining and using, 127–129
  - DisplayRatio(), 152–153
  - DisplayRounded
    - Decimal(), 141
  - implementing default arguments, 140–142
  - initialization, calling, 233
  - InputInterestData(), 132, 133
  - InputPositive
    - Decimal(), 133–134
  - IsTerminateString(), 192–194
  - matching argument definitions with usage, 138–139
  - mathematical, 64
  - member, 128
  - mixing with methods, 174–176
  - naming, 129, 139–140, 192
  - with no value, defining, 152–153
  - nonstatic member, invoking, 167–169
  - nonvoid, 152
  - OutputInterestData(), 132–133
  - overloading, 139–140, 141–142, 243
  - parentheses and, 128
  - passing argument to, 136
  - passing multiple arguments to, 136–138
  - passing object to, 163–165
  - passing value-type arguments, 142–147
  - passing variable as out argument to, 372–373
  - planning, 130
  - refactoring and, 129, 130
  - RemoveSpecialChars(), 204–205
  - returning value to caller, 147–148, 149
  - returning value using pass by reference, 148–149
  - SetName(), 164, 166, 168
  - sin(), 147
  - Sort(), 125
  - static member, defining, 165–167
  - TestString(), 150–151
  - Update(), 145–146
  - user of, 140
  - variables within, 146–147
  - void, 152
  - word processor and, 135
  - WriteLine(), 160

## • G •

- garbage collector, 108, 272
- generic class, methods in, 353–355, 356
- generic code, writing, 348–349
- generic collections
  - as efficient, 337
  - list of, 338
  - List<T>, 338–340
  - priority queue, 341–345
  - <T> and, 338
  - as type-safe, 336–337
  - writing, 340–341
- generic factory, building
  - creating objects generically, 362
  - default constructor and, 363
  - GenericInterface
    - program, 359–361
  - one-parameter construction, 362–364
- generic interface, 357–358
- generic iterator, CD91
- generic method
  - constraints and, 356
  - in generic class, 356
  - in nongeneric class, 355
- GenericCollections
  - program, 338–340
- GenericInterface
  - program, 359–361

GenericMethod program, 354–355  
 generics feature, 2, 333, 381  
 { get; } notation, 346  
 GetBalance() method, 226–227  
 GetValue() method, 319  
 GNU software, 388  
 goto statement, 98  
 graphical Windows  
   application, writing, 3

## • H •

HAS\_A relationship, 259–261  
 heap object, 380  
 Help Favorites, CD124  
 Help system, CD119–CD124  
 Help→Index, 177  
 hexadecimal numbers, CD59  
 hiding  
   base class method, 275–280, 374–375  
   window, CD103  
 HidingWithdrawal  
   program, 277–278  
 HidingWithdrawal  
   Polymorphically  
   program, 282–283  
 hierarchy of inherited  
   classes, 266  
 high-level computer  
   language, 12  
 How Do I button, CD124  
 Hungarian notation, 54

## • I •

icon  
   Console Application, 30  
   lightning bolt, 25  
 if statement  
   else statement and, 75–77  
   embedded, 77–79  
   overview of, 72–75  
 immutability of string, 188–189  
 implementing interface, 305, 376  
 implicit type conversion, 67–68, 370–371  
 include file, 382  
 increment operator, 61–62

incrementing counting  
   variable in while loop, 83–84  
 indentation, using for  
   clarity, 75  
 index and array, 113  
 Index Help, CD120–CD122  
 indexer, CD76–CD80  
 Indexer program, CD77–CD80  
 IndexOf() method, 204–205  
 infinite loop, 84, 89, 92  
 inheritance. *See also*  
   inheriting  
     as operator and, 264–265  
     BankAccount class and, 254–257  
     changing class of object  
       and, 261–262  
     class and, 252–253  
     ConstructorSavings  
       Account program, 269–271  
     default base class  
       constructor, invoking, 265–266  
     description of, 251  
     destructor and, 271–272  
     HAS\_A relationship and, 259–261  
     hierarchy, restarting, 296–299  
     importance of, 253–254  
     interface and, 316–319  
     invalid cast and, 262–263  
     is operator and, 263–264  
     IS\_A property and, 257–258, 260–261  
     polymorphism and, 284  
     power of, 273–274  
     problems of, 289–291  
 InheritanceExample  
   program, 252–253  
 InheritanceTest program, 296–299  
 inherited method,  
   overloading, 274–275  
 inheriting. *See also*  
   inheritance  
     base class polymorphically, 374–375  
     from multiple base classes, in C++, 383

InheritingAConstructor  
   program, 265–266  
 Init() method, 172  
 initialization function,  
   calling, 233  
 initializing  
   data member, 172  
   object directly, 241–242  
   structure, 322  
   variable, 41–42  
 InputInterestData()  
   function, 132, 133  
 input/output (I/O), CD43  
 InputPositiveDecimal()  
   function, 133–134  
 installing items from CD to  
   hard drive, 386–387  
 instance, 103, 167, 216  
 instance method, 168  
 instantiating class locally, 249  
 int variable  
   description of, 41–42  
   fractions and, 43–44  
   Int32 structure and, 327, 328–330  
   types of, 42–43  
 integer, 41  
 integer truncation, 44  
 integer variable types, 42–43  
 Intel processor  
   Pentium, and floating point  
   numbers, 47  
   registers and, 52  
 interest, calculating, 73  
 interface  
   C++ and, 383–384  
   creating, 308–309  
   description of, 305–306  
   example of, 306–307  
   flexibility and, 315  
   implementing, 305, 376  
   inheriting, 316–319  
   nongeneric versus generic, 357–358  
   predefined, 309–310  
   usability of, 217–218  
 interface keyword, 305  
 internal member of  
   class, 225  
 internal protected  
   member of class, 225  
 intrinsic variable type, 52  
 Int32 structure, 327, 328–330

- InvokeBaseConstructor program, 268–269
  - InvokeMethod program, 167–168
  - I/O (input/output), CD43
  - IPrioritizable interface, 346
  - is keyword, 285
  - is operator, 263–264, 337
  - IS\_A property, 253, 257–258, 260–261
  - IsAllDigits program, 196–198
  - IsTerminateString() function, 192–194
  - iterating
    - days of month, CD84–CD85
    - foreach collections, CD72–CD75
    - through array, 192
    - through containers, 120–121
    - through directory of files, CD55–CD61
  - iterator block
    - description of, 2
    - looping around, CD80–CD84
    - placement of, CD92–CD98
    - types of, CD88–CD91
  - iterator syntax, CD87–CD88
  - IteratorBlockIterator program, CD95–CD98
  - IteratorBlocks program, CD81–CD84
- **J** •
- Java programming language, 13–14, 181, 379
- **K** •
- key, CD78
  - keyboard shortcuts
    - Alt+Tab (switch program), CD129
    - Ctrl+C (interrupt running program), CD56
    - Ctrl+F5 (Start Without Debugging), 159
    - Ctrl+K then Ctrl+X (view menu of snippets), 34
    - Ctrl-Space (display auto-complete), 26
    - F11 (Step Into), 238, CD130
    - F5 (Start Debugging), 159
    - F1 (Help), CD119–CD120
    - F10 (Step Over), CD128
    - Shift+F5 (Stop Debugging menu), CD137
  - keywords
    - as operator, 264–265
    - base, 268–269, 280–281
    - interface, 305
    - is, 263–264, 285
    - new, 280
    - null, 353
    - out, 144, 145, 146, 148–149
    - override, 286–287
    - private, 222, 224, 225
    - ref, 144, 145, 148–149
    - sealed, 300
    - static, 110
    - string, 187
    - this, 171–176
    - virtual, 286–287
    - void, 152
- **L** •
- late binding, 283
  - Length property of array, 117
  - level of abstraction, 214
  - levels of security, 224–225
  - lightning bolt icon, 25
  - linked list of objects, 234, CD61–CD71
  - LinkedList class, 334
  - LinkedListContainer program, CD62–CD71
  - LinkedListWithIterator Block program, CD92–CD95
  - List<T> generic collection, 338–340
  - Locals window, 239
  - logical comparison operators
    - compound, 64–66
    - floating point numbers and, 63–64
    - overview of, 62–63
  - long variable, 42
  - looping around iterator block, CD80–CD84
  - looping statements
    - break and continue commands and, 84–86
    - description of, 80
    - do...while, 84
    - for, 90–92
    - foreach command, 120–121
    - infinite, 84, 89, 92
    - nested, 92–95
    - scope rules, 89–90
    - while, 80–84, 86–89
  - LoopThroughFiles program, CD55–CD61
- **M** •
- machine language, 11–12
  - Main() function
    - Indexer class, CD79
    - overview of, 153–154
    - passing argument from DOS prompt, 155–157
    - passing argument from Visual Studio 2005, 159–162
    - passing argument from window, 157–159
  - PassObjectToMember Function program and, 166
  - for PriorityQueue program, 347–348
  - Main() method, 34–35
  - manually controlling output
    - Format() method, 206–210
    - overview of, 200–201
    - Trim() and Pad() methods, 201–203
  - mathematical functions, 64
  - member function, 128
  - member of class
    - description of, 102–103
    - non-static, 128
    - public member, 221–224, 225
    - static, 110
  - member of object, accessing, 104–106
  - memory allocation in C++, 380
  - memory block, unreachable, 272

- 'methodName': not all code paths return a value error message, 376–377
  - methods
    - abstract, 296
    - accessor, 226–227, 231
    - auto-complete feature and, 179–180
    - base class, overloading, 275–280, 374–375
    - class and, 163, 168
    - Compare(), 189–193
    - Concat(), 204–205
    - creating and renaming, 26
    - current object and, 169–176
    - Decimal.Round(), 228–230
    - defining, 167–169
    - definition of, 128, 168
    - Dequeue() method, 342, 350
    - Enqueue(), 342, 349–350
    - expanding full name, 168–169
    - external, 224
    - Format(), 206–210
    - generic class and, 353–355, 356
    - GetBalance(), 226–227
    - GetValue(), 319
    - IndexOf(), 204–205
    - inherited, overloading, 274–281
    - Init(), 172
    - instance, 168
    - Main(), 34–35
    - mixing with functions, 174–176
    - name of, 168–169, 274–275
    - namespace and, CD33
    - nongeneric class and, 355
    - object and, 168
    - overloading in base class, 374–375
    - Pad(), 201–203
    - renaming, 26
    - Replace(), 203–205
    - Split(), 198–200, 205–206
    - of structure, 323
    - Swap(), 353
    - ToInt32(), 196
    - TopQueue(), 350–351
    - ToString(), 323, CD26
    - Trim(), 196, 201–203
    - writing, CD118
  - Microsoft
    - Visual Basic, 12, 28
    - Visual Basic .NET, 379–380
    - Visual C# 2005, 385
  - Microsoft C++ programming language
    - constructors and, 382–383
    - global data or functions and, 380
    - interface and, 383–384
    - memory allocation and, 380
    - multiple inheriting and, 383
    - overview of, 379–380
    - pointer variables and, 381
    - template feature, 381–382
    - type checking and, 382
    - Unified Type System, 384
    - variable types and, 383
  - Microsoft .NET Framework C# and, 13
    - Forms Designer and, 28
    - languages supported by, 14
    - overview of, 13–14, 379–380
    - tools, CD140–CD142
  - Microsoft Visual Studio. *See* Visual Studio
  - Microsoft Visual Studio 2005. *See* Visual Studio 2005
  - minus (-) sign, 31
  - misspelling variable name, 368
  - MixingFunctionsAndMethods program, 174–176
  - MixingFunctionsAndMethodsWithXMLTags program, 181–184
  - modifying string, 188–189
  - ModifyString program, 188–189
  - module, 225
  - modulo (%) operator, 58, CD60
  - Mono implementation, 1
  - multiplication operator, forms of, 67
  - MyException program, CD17–CD20, CD22
- N ●
- The name 'memberName' does not exist in the class or namespace 'className' error message, 368–369
  - name of method, 168–169, 274–275
  - named iterator, CD89–CD90
  - namespace, CD29, CD30–CD38
  - NamespaceUse program, CD36–CD38
  - naming
    - array, 120
    - class, 102
    - constant, 111, CD6
    - destructor, 271
    - file or directory, 156
    - function, 129, 139–140, 192
    - interface, 305, 306
    - object reference variable, 120
    - project, 18
    - property, 231
  - naming conventions, 54, CD117–CD118
  - navigating code, CD110
  - nested loop, 92–95
  - nested statement, 77
  - .NET Framework C# and, 13
    - Forms Designer and, 28
    - languages supported by, 14
    - overview of, 13–14, 379–380
    - tools, CD140–CD142
  - .NET package, downloading, 3
  - .NET redistributable package, CD180
  - netable language, 299
  - new() constraint, 362
  - new keyword, 280
  - New Project dialog box, 16, 17
  - newline character, 106
  - nondeterministic destruction, 272

- nongeneric collections
    - boxing and unboxing, 336, 337
    - list of, 334–335
    - methods and, 355
    - overview of, 334
    - using, 335–336
  - nongeneric interface, 357–358
  - `NongenericCollections` program, 335–336
  - nonstatic member function, invoking, 167–169. *See also* methods
  - nonvoid function, 152
  - notation, Hungarian, 54
  - notational C#, 190
  - null keyword and generic class, 353
  - null object, 107
  - null reference, 150–151
  - null string, 52, 150
  - numbers
    - hexadecimal, CD59
    - series of, handling, 198–200
  - numeric constant, declaring, 54–55
  - numeric input, parsing, 196–198
  - numeric types, logical comparisons for, 63
  - NUnit testing tool
    - debugging test code, CD172–CD175
    - description of, 6, 389
    - running, CD164–CD165
    - unit testing and, CD165–CD166
    - writing test class, CD166–CD172
- 0 •
- object
    - accessing members of, 104–106
    - adding to linked list, CD67–CD68
    - arrays and, 118–120
    - in C++, 380
    - changing class of, 261–262
    - description of, 103–104, 233
    - discriminating between numerous, 106
    - initializing directly, 241–242
    - linked list of, 234
    - method and, 168
    - passing by reference, 163–165
    - reachable, 108
    - removing from linked list, CD68–CD69
    - structure object, 320–322
    - unreachable, 108
  - `Object` class, 327, 328, 330
  - object class, 264
  - object property, 110
  - object reference variable, naming, 120
  - object-based language, 284
  - object-oriented programming. *See also* inheritance
    - abstraction, 213–215
    - access control, 218–219
    - C++ and, 380
    - C# support for, 219
    - classification, 215–217
    - polymorphism and, 284
    - usable interface, 217–218
  - opening
    - DOS window, 33
    - Output window, 18
    - Solution Explorer, 30, 160
    - Toolbox, 21
  - open-source software, 1
  - operating orders of
    - arithmetic operators, 58–59
  - operation, calculating type of, 67–68
  - out keyword, 144, 145, 146, 148–149
  - output, controlling manually
    - `Format()` method, 206–210
    - overview of, 200–201
    - `Trim()` and `Pad()` methods, 201–203
  - Output window, 18, 19
  - `OutputFormatControls` program, 208–209
  - `OutputInterestData()` function, 132–133
  - `OutputName()` function, 164, 165, 166
  - overloading
    - base class method, 275–280
    - constructor, 243–245
    - function, 139–140, 141–142, 243
    - inherited method, 274–281
    - method in base class, 374–375
  - override keyword, 286–287
  - overriding `Exception` class, CD22–CD26
- p •
- `PackageFactory` program, 358–359
  - `Pad()` method, 201–203
  - parameter
    - auto-complete feature and, 178–179
    - implementing default, 140–142
    - matching definitions with usage, 138–139
    - multiple, passing to function, 136–138
  - as part of name of function, 274–275
  - passing from DOS prompt, 155–157
  - passing from Visual Studio 2005, 159–162
  - passing from window, 157–159
  - passing to default base class constructor, 266–269
  - passing to function, 136
  - value-type, passing by reference, 143–147
  - value-type, passing by value, 142–143
- parentheses
  - cast operator and, 68, 69
  - functions and, 128
  - order of precedence and, 59
- `ParseSequenceWithSplit` program, 198–200

- parsing
  - characters out of string, 194–196
  - numeric input, 196–198
- partial class, CD177–CD178
- PassByReference program, 143–144
- PassByReferenceError program, 145–146
- PassByValue program, 142–143
- passing. *See also* passing argument
  - current object, 169–171
  - object by reference, 163–165
  - variable as out argument to function, 372–373
- passing argument
  - to default base class constructor, 266–269
  - from DOS prompt, 155–157
  - to function, 136
  - multiple, to function, 136–138
  - value-type, by reference, 143–147
  - value-type, by value, 142–143
  - from Visual Studio 2005, 159–162
  - from window, 157–159
- PassObject program, 163–164
- PassObjectToMember Function program, 165–166
- path, CD49
- PDP-8 computer, 61
- Pentium processor and floating point numbers, 47
- pipe, double (||) operator, 66
- pipe (!) operator, 65
- plus (+) operator, strings and, 52
- plus (+) sign, code region and, 31
- pointer, 380–381
- pointer variable, 381
- PolymorphicInheritance program, 286–287
- polymorphism
  - declared type and, 283–284
  - description of, 219, 251, 283
  - example of, 282–283
  - is keyword and, 285
  - virtual keyword and, 286–287
- Portable .NET
  - implementation, 1
- postincrement operator, 61, 62, 92
- predefined interface, 309–310
- preincrement operator, 62
- preventing confusion in code, 59, 60
- Preview Code Changes pane, 26
- PriorityQueue class. *See also* PriorityQueue program
  - constraints for, 351–352
  - Dequeue() method, 350
  - Enqueue() method, 349–350
  - members of, 351
  - null value for type T and, 352–353
  - TopQueue() utility method, 350–351
  - underlying queues, 349
- PriorityQueue program
  - code for, 342–345
  - IPrioritizable interface, 346
  - Main() function for, 347–348
  - Package class, 345–346
- private keyword, 222, 224, 225
- process, CD173
- processor upchuck, 153
- program. *See also* console application; *specific programs*
  - action, adding, 25–27
  - breaking, CD132–CD135
  - building and running, 18–20
  - console, creating, 29–31
  - converting class into, CD114–CD115
  - creating, 15
  - description of, 12
  - developing, CD142
  - dividing into multiple assemblies, CD29–CD30
  - dividing into multiple source files, CD28–CD29
  - executable, 12, 373
  - executing, 19, 32, 35
  - Forms Designer and, 20–24
  - freeware, 388
  - rebuilding and running, 24–25, 373–374
  - running on different machines, CD180
  - shareware, 388
  - source files for example, 387–388
  - template, creating, 15–18
  - testing, 27–28
- project
  - creating, 30
  - description of, 16
  - displaying, CD108–CD110
  - naming, 18
  - properties of, accessing, 160–161
  - startup, CD30
- project file, CD29, CD107
- Project→Properties, 279
- promotion, 67–68
- properties
  - of control, 23–24
  - inheritance and, 251
  - in interface declaration, 346
  - IS\_A, 253
  - of project, accessing, 160–161, CD109–CD110
- Properties window, 23
- protected member of class, 225
- protection level, specifying, 371–372
- pseudocode, 190
- public member of class, 221–224, 225
- public modifier, 103

## • Q •

Queue class, 334  
 queue data structure  
 description of, 341  
 PriorityQueue program,  
 342–345  
 rules for, 341–342  
 quotation marks, string  
 variable compared to  
 char variable, 53

## • R •

reachable object, 108  
 reading comments, 129  
 ReadLine() command, 74  
 real number, 44  
 real type, 283  
 rearranging windows,  
 CD103–CD104  
 rebuilding program, 24–25,  
 373–374  
 recursing, 280  
 redundancy, reducing, 292  
 ref keyword, 144, 145,  
 148–149  
 Refactor menu (Visual Studio  
 2005), 26, 133  
 refactoring, 129, 130, 133  
 Refactor→Rename, 26  
 reference, passing value-type  
 argument by, 143–147  
 reference type variable  
 boxing, 330–331  
 description of, 53, 150–151,  
 320  
 operators defined on,  
 107–108  
 ReferencingThis  
 Explicitly program,  
 172–174  
 region, adding, 31  
 registers, 52  
 RemoveSpecialChars()  
 function, 204–205  
 RemoveWhiteSpace  
 program, 204–205

RemoveWhiteSpaceWith  
 Split program, 205–206  
 removing  
 breakpoint, CD134  
 character from end of  
 string, 201  
 object from linked list,  
 CD68–CD69  
 renaming method, 26  
 Replace() method, 203–205  
 resizing text box, 22  
 resources on C#. *See also*  
 Web sites  
 Brandon Bray's blog, 382  
 browsing online help, 177  
 Web sites, 7–8  
 restarting class hierarchy,  
 296–299  
 rethrowing error, CD20–CD21  
 return statement, 147–148,  
 149  
 returning error  
 overview of, CD1–CD3  
 problems with, CD7–CD8  
 returning value, 147–149  
 reuse, 254  
 rounding, 44  
 run-time error  
 catch blocks, assigning  
 multiple, CD15–CD17  
 description of, 262, CD126  
 error codes and, CD4–CD7  
 exception mechanism for,  
 CD1  
 returning, CD1–CD3,  
 CD7–CD8  
 run-time type, 283

## • S •

Save button, 24  
 scope of variable, 89–90, 369  
 sealed keyword, 300  
 sealing class, 375  
 Search facility, 155  
 Search Help, CD122–CD123  
 Search Results window, 156  
 security, levels of, 224–225  
 series of numbers, handling,  
 198–200

SetName() function, 164,  
 166, 168  
 setup project, CD180  
 shareware program, 388  
 SharpDevelop program, 6,  
 385, 389, CD142–CD149  
 Shift+F5 (Stop Debugging  
 menu), CD137  
 short-circuit evaluation, 66  
 side effect of property, 232  
 signed integer variable, 43  
 Simple Factory class,  
 358–359  
 simple operators, 57–58  
 simple value type, 327  
 SimpleSavingsAccount  
 program, 254–257  
 sin() function, 147  
 single stepping,  
 CD128–CD131  
 smart tag, 22  
 snaplines, 22  
 snippets, 34–35  
 software  
 elegance in, 261  
 GNU, 388  
 open-source, 1  
 trial, demo, or evaluation  
 versions of, 388  
 solution, CD29, CD107  
 Solution Explorer, 19, 30, 160,  
 CD106–CD115  
 Sort() function, 125  
 sorting arrays of objects,  
 122–126  
 SortInterface program  
 code for, 311–314  
 explanation of, 314–315  
 interface, creating, 308–309  
 output of, 315–316  
 overview of, 307  
 predefined interface,  
 309–310  
 SortStudents program,  
 123–124  
 source file  
 collecting into namespace,  
 CD30–CD38  
 description of, 12, 387–388  
 dividing program into  
 multiple, CD28–CD29

- source program, creating, 30–31
  - Southern Naming Convention, CD6
  - special characters, 50–51
  - Split() method, 198–200, 205–206
  - Stack class, 335
  - stack object, 380
  - stack trace, CD13, CD135–CD136
  - stacking windows, CD104–CD106
  - starting
    - console program by double-clicking, 157–158
    - constructor from debugger, 238–241
    - program, 19, 32, 35
    - Visual Studio, 16
  - Start→Programs→Accessories→Command Prompt, 155
  - startup project, CD30
  - static keyword, 110
  - static member function, defining, 165–167
  - static property, 232
  - stepping over, CD129
  - StreamReader class, CD50–CD54
  - StreamWriter class, CD45–CD50
  - string
    - addition operator and, 188
    - Compare() method and, 189–193
    - empty, 52, 150
    - Format() method, 206–210
    - modifying, 188–189
    - null, 52, 150
    - numeric input, parsing, 196–198
    - parsing characters out of, 194–196
    - Replace() method, 203–205
    - series of numbers, handling, 198–200
    - Split() method, 205–206
    - switch() control and, 193–194
    - Trim() and Pad() methods, 201–203
    - white space and, 195
  - String class, 187
  - string constant, 189
  - string keyword, 187
  - string objects, 153
  - string type, as reference type, 328
  - string variable, 51–52, 53
  - StringToCharAccess program, 194–195
  - structure
    - class compared to, 327
    - description of, 320–322, 384
    - features of, 323–326
    - methods of, 323
    - predefined types, 327–328
    - type unification and, 328–330
  - structure constructor, 322–323
  - StructureExample program, 324–326
  - subclass, 216, 274
  - ‘subclassName’: cannot inherit from sealed class ‘baseclassName’ error message, 375
  - ‘subclassName.methodName’ hides inherited member ‘baseclassName.methodName.’ Use the new keyword if hiding was intended error message, 374–375
  - subexpression, 67
  - superclass, extracting, 291
  - Swap() method, 353
  - switch, 154
  - switch control, 96–98, 193–194
  - syntax for structure declaration, 320
  - syntax of arrays, 117–118
  - System Library, 177–179
  - system requirements for CD-ROM, 385–386
  - System.IO namespace, CD43, CD44
- T ●
- <T> and generic collections, 338
  - tabbed window, CD102
  - template, creating, 15–18, 29–31
  - template, in C++, 381–382
  - Templates pane, Console Application icon, 30
  - terminating program, 20, 373–374
  - ternary operator, 69–70
  - testing
    - after internal changes to class, 230
    - code on different machines, CD180
    - console application, 31
    - NUnit tool, 389
    - variable for different values, 96–98
  - TestString() function, 150–151
  - Textbox control, 22, 23
  - TextPad editor
    - Build C# Debug tool, adding, CD153–CD155
    - C# document class, creating, CD152–CD153
    - compiler errors and, CD160–CD161
    - overview of, 6, 389, CD149–CD152
    - Parameters fields items, CD156–CD160
    - Release build tool, configuring, CD156
    - tools configuration options, CD151
    - tools, configuring, CD161–CD164
  - this keyword, 171–174
  - three-slash (///) comment, 181, CD117
  - throwing exception, CD17–CD19
  - tight coupling, 359
  - tilde (~), 271
  - ToDecimal() command, 74

ToInt32() method, 196  
 toolbar in Forms Designer, 19  
 Toolbox, controls in, 21–22  
 Tools→Command Window, 33  
 Tools→Options→Projects  
     and Solutions→  
         General, 18  
 TopQueue() utility method,  
     350–351  
 ToString() method,  
     323, 330, CD26  
 Trim() method, 196,  
     201–203  
 troubleshooting. *See also*  
     error message  
     CD-ROM, 389–390  
     fixed-value array and, 113  
     overview of, 367  
 truncation, 44  
 try...catch statement, 210  
 type-safety, 336–337, 381  
 TypeUnification program,  
     328–330

## • U •

uint variable, 43  
 UML (Unified Modeling  
     Language), 288, 290,  
     CD106  
 Unable to copy the file  
     ‘programName.exe’ to  
     ‘programName.exe.’ The  
     process cannot... error  
     message, 373–374  
 unary negative operator, 58  
 unboxing  
     nongeneric collections and,  
         336, 337  
     reference type variable and,  
         331  
 Unified Modeling Language  
     (UML), 288, 290, CD106  
 Unified Type System, 384  
 uninitialized reference, 107  
 Unix implementations, 1  
 unreachable object, 108  
 unsigned integer variable, 43  
 up conversion, 68  
 Update() function, 145–146

Use of unassigned local  
     variable ‘n’ error  
     message, 372–373  
 user of function, 140

## • V •

value  
     defining function with no,  
         152–153  
     multiple, returning from  
         single function, 149  
     passing value-type  
         argument by, 142–143  
     returning to caller, 147–148,  
         149  
     returning using pass by  
         reference, 148–149  
 value type variable  
     boxing, 149, 330–331  
     description of, 52–53, 320  
 variable  
     bool type, 49, 64–66  
     changing type of, 55–56  
     char type, 50–51, 53  
     class compared to  
         member, 128  
     counting, in for loop, 92  
     counting, incrementing in  
         while loop, 83–84  
     decimal type, 48–49  
     declaring, 40, 42, 319–320,  
         368, 369  
     description of, 39  
     double, 45  
     float, 45, 47  
     floating point, 44–49  
     within function, 146–147  
     initializing, 41–42  
     int, 41–44, 327, 328–330  
     integer types, 42–43  
     intrinsic type, 52  
     long, 42  
     name of, 54, 368  
     object reference,  
         naming, 120  
     passing as out argument to  
         function, 372–373  
     pointer, 381  
     reference, 53, 150–151, 320,  
         330–331

    scope of, 89–90  
     signed integer, 43  
     string compared to  
         char, 53  
     string type, 51–52, 53  
     testing for different values,  
         96–98  
     types of, 369–371, 383  
     uint, 43  
     unsigned integer, 43  
     value-type, 52–53, 142, 320,  
         330–331  
 VariableArrayAverage  
     program, 115–116, 117  
 variable-length array,  
     114–118  
 VehicleDataOnly program,  
     104–105  
 View menu, CD100, CD101  
 View→Properties Window, 23  
 View→Solution Explorer, 160  
 View→Toolbox, 21  
 virtual keyword, 286–287  
 Visual Basic, 12, 28  
 Visual Basic .NET, 379–380  
 Visual C# 2005, 385  
 Visual Studio  
     Application Wizard, 16, 17,  
         CD32  
     auto-complete feature,  
         176–184  
     generating XML  
         documentation with, 185  
     history of, 14–15  
     starting, 16  
     versions of, 1, 3  
 Visual Studio 2005  
     CD-ROM and, 385  
     class, adding, CD110–CD111  
     Class Designer, CD106  
     customizing window layout,  
         CD100–CD106  
     DataTip, CD134–CD135  
     debugging tools,  
         CD126–CD138  
     Help system, CD119–CD125  
     overview of, CD99–CD100,  
         CD139  
     passing argument from,  
         159–162  
     Refactor menu, 26, 133

- Visual Studio 2005 (*continued*)
    - SharpDevelop compared to, CD143–CD144
    - Solution Explorer, 19, 30, 160, CD106–CD115
  - void function, 152
  - void keyword, 152
  - VSDebug program, CD126–CD127
  - VSInterface program, CD112–CD114, CD115
- **W** •
- warning. *See also* error message
    - fixed-value array and, 113
    - fixing, 279
    - overview of, 367
    - studying, CD127–CD128
  - Web Services and .NET, 14
  - Web sites
    - author, 8, 390
    - DotGNU, 1
    - languages supported by .NET, 14
    - Mono, 1
    - .NET Framework tools, CD140
    - .NET package, 3
  - NUnit testing tool, CD164
  - refactoring, 133
  - resources on C#, 7–8
  - SharpDevelop program, CD142
  - Wiley Product Technical Support, 390
- while loop
  - break statement and, 86–89
  - common mistakes in, 92
  - do...while loop, 84
  - overview of, 80–84
 white space, 195
  - Wiley Product Technical Support, 390
 window
  - layout, customizing, CD100–CD106
  - passing argument from, 157–159
 Windows Forms (WinForms), 15, 18–20
  - word processor, functions for, 135
 wrapper class, 342
  - WriteLine() command, as function call, 74, 160
 writing
  - code, CD115–CD119
  - collection class, CD61–CD71
  - constant, CD6
  - forms code, CD175–CD179
  - generic code, 348–349
  - generic collections, 340–341
  - graphical Windows application, 3
  - method, CD118
  - test class for NUnit testing tool, CD166–CD172

• **X** •

  - xcopy deployment, CD180
  - XML (eXtensible Markup Language), 180–185

• **Y** •

  - yield break statement, CD88
  - yield return statement, CD87–CD88

• **Z** •

  - zero reference, 150–151
  - zero, trailing, showing, 83