

Index

COPYRIGHTED MATERIAL

Index

SYMBOLS

:: (colons), scope resolution operator

- classes, 341–342
- definition, 26
- example, 93–94

, (comma), expression separator, 76–77

. (period)

- member access operator, 325
- member selection operator, 325

; (semicolon), end of statement indicator, 47

& (ampersand)

- AND, 82–83
- address-of operator, 173–174

&& (ampersands), logical AND, 125

>> (angle brackets), extraction operator, 62–63

-> (arrow)

- indirect member access operator, 332
- indirect member selection operator, 332

* (asterisk), indirection operator, 174

^ (caret), exclusive OR, 85

“...” (double quotes), special character indicator, 65

= (equal sign), assignment operator, overloading, 418–422

== (equal signs), relational operator, 115–116

! (exclamation mark), logical NOT, 126

!= (exclamation mark, equal sign), relational operator, 115–116

> (greater than)

- operator overloading, 415–417
- OR, 84–85
- relational operator, 115–116

>= (greater than, equal sign), relational operator, 115–116

<< (left angle brackets), shift left, 86–87

< (less than), relational operator, 115–116

<= (less than, equal sign), relational operator, 115–116

-> (minus sign, greater than)

- indirect member access operator, 332
- indirect member selection operator, 332

-- (minus signs), decrement operator

- description, 74–76
- overloading, 426–427, 467

+ (plus sign)

- addition operator, overloading, 423–426
- join operator, 217

++ (plus signs), increment operator

- description, 74–76
- overloading, 426–427, 467

> (right angle bracket), OR, 84–85

>> (right angle brackets), shift right, 86–87

// (slashes), comment indicator, 44

|| (vertical bars), logical OR, 125–126

{ } (curly braces), function syntax, 46

~ (tilde), NOT, 86

A

abstract classes, inheritance, 505–508

access control

- classes, 336
- inheritance, 479–482, 489–490

access specifiers, 531–532

Add() **function**, 1094

AddCopy() **function**, 1094

AddHead() **function**, 764–766, 772

AddNew() **function**, 980–982, 1121

addresses, returning values from, 256–257

AddTail() **function**, 764–766, 772

AFX_EXT_CLASS **keyword**, 912–913, 916

aliases. See references; tracking references.

alignment, cells, 1100

Alignment **property**, 1100

allocating memory. See dynamic memory allocation.

AlternatingRowsDefaultCellStyle **property**, 1101

ampersand (&)

AND, 82–83

address-of operator, 173–174

ampersands (&&), logical AND, 125

angle brackets (>>), extraction operator, 62–63

anonymous unions, 409–410

API (application programming interface), 617–618

application class, 653

application programming interface (API), 617–618

Application wizard. See MFC Application wizard.

Arc() **function**, 716–717

arguments, to functions

arrays as, 243–247

arrays of function pointers as, 277

definition, 232

function pointers as, 275–277

passing to main(), 250–252, 267–268

passing-by-pointer, 241–243

passing-by-reference, 247–249

passing-by-value, 240–241

pointer notation, 244–245

variable number of, 252–254, 266–267

arithmetic operations

, (comma), expression separator, 76–77

-- (minus signs), decrement operator, 74–76

++ (plus signs), increment operator, 74–76

assignment statements, 67–68

basic operators, 68

bitwise operators

& (ampersand), AND, 82–83

^ (caret), exclusive OR, 85

<< (left angle brackets), shift left, 86–87

> (right angle bracket), OR, 84–85

>> (right angle brackets), shift right, 86–87

~ (tilde), NOT, 86

description, 81

inclusive OR, 84–85

casting types

in assignment statements, 79–80

C++/CLI, 108–109

definition, 78

explicit casts, 80–81

old-style casts, 81

rules for, 78–79

safe_cast operation, 108–109

const modifier, 70–71

example, 69–73

expressions, separating, 76–77

Lvalues, 68

memory addresses, 68

modifying variables, 73–74

operator precedence, 77–78

with pointers, 185–187

remainders, calculating, 69–73

Rvalues, 68

arrays

of arrays, 213–216

character, 166–169

CLR, 200–205

collection classes

CArray template class, 761–763

collections of objects, 761–763

definition, 760

collections of objects, 761–763

elements, 160

entering data, 163

as function arguments, 243–247

indexing, 160–161

multidimensional

basic structure of, 169–170

initializing, 170–172

passing to functions, 245–247

pointer notation, 191–192

string handling, 171–172

names, as pointers, 187–189

null character, 166

of objects, 363–364

one-dimensional

basic structure of, 160–161

declaring, 161–164

initializing, 164–165

string handling, 166–169

string termination character, 166

and pointers, 185–192

of pointers, 179–181

size of, determining, 181–183

sizeof operator, 181–183

arrow (->)

indirect member access operator, 332
indirect member selection operator, 332

ASCII codes, table of, 1133–1137**assemblies, 25****Assert () function, 578–580, 607–608**

assertions, 578–580, 607–608

assignment statements, 47, 67–68, 79–80**asterisk (*), indirection operator, 174****automatic variables, 88–91****B****b prefix, 619–620****BackColor property, 1099–1100****background color, cells, 1099–1100****base classes**

accessing private members, 480–482
definition, 475–476
deriving classes from, 476–479
direct, 475–476
indirect, 475–476, 508–511
pointers to, 501–503

BeginTrans () function, 983–984**binding controls to**

columns, 1123–1127
data sources, 1115–1120

BindingNavigator control, 1120–1123**BindingSource component, 1115–1120****bitwise operators**

& (ampersand), AND, 82–83
^ (caret), exclusive OR, 85
<< (left angle brackets), shift left, 86–87
> (right angle bracket), OR, 84–85
>> (right angle brackets), shift right, 86–87
~ (tilde), NOT, 86
description, 81
inclusive OR, 84–85

black-and-white drawing, 747**block scope, 88****body, function, 234–235****bool data type, 57****BOOLEAN data type, 618****Boolean variables, 57****borders, windows, 615****bound mode, 1093, 1114–1115****bounding rectangles, 739–741****boxing data types, 101****branching, unconditional, 132****breakpoints, 570–572****Browser style toolbar option, 659****brushes, 719–721****bugs. See also debugging.**

common examples, 567–568
definition, 565
semantic errors, 567
syntactic errors, 567

Button control, 1044–1046**buttons**

cells, 1095
in cells, 1095
clicks, handling, 1076–1079, 1083–1086
definition, 819
illustration, 819
labels, changing, 992–993
movement, detecting, 725

by prefix, 619–620**BYTE data type, 618****C****c prefix, 619–620****C++ programs. See also C++/CLI programs; program structure.**

basic development options, 3–4
managed C++, 3
native C++, 2, 3
simple example, 39–44
standards, 5–6
unmanaged C++, 3

C++/CLI programs. See also specific topics.

definition, 5–6
overview, 150–156

calculations. See arithmetic operations.**calculator example**

analyzing a number, 299–302, 318–319
C++/CLI, 315–320
expressions, evaluating, 295–298, 316–317
extending the program, 304–305
project description, 291–294
returning values from terms, 298–299, 318
running the program, 307–308
source code, 302–303
strings
extracting substrings, 305–307, 319–320
removing blanks, 294–295, 316

call stack, debugging, 585–587

calling functions

- counting calls, 260
- process description, 235–239
- recursively, 262–265

Cancel button, visibility, 993–994

canceling database update, 980–982, 982–984, 997–998

CancelUpdate() function, 980–982

CArchive class, serialization, 870–871

caret (^), exclusive OR, 85

CArray template class, 761–763

case labels, 130

casting

types

- in assignment statements, 79–80
- C++/CLI, 108–109
- definition, 78
- explicit casts, 80–81
- old-style casts, 81
- rules for, 78–79
- safe_cast operation, 108–109

variables

- in assignment statements, 79–80
- definition, 78
- explicit casts, 80–81
- old-style casts, 81
- rules for, 78–79

catch blocks, 281–282

catching exceptions

- C++/CLI, 308–309
- example, 280–281

CBox class

- analyzing objects, 441–445
- class implementation, 444
- combining objects, 439–441
- comparing objects, 438–439
- defining
 - the constructor, 447–448
 - data members, 446–447
 - function members, 448–453
 - global functions, 453–455
- source code, 437–438
- using, 455–458

CCircle class, 744–746

CCurve class

- defining, 775–777
- drawing curves, 746
- exercising, 778–779
- implementing, 777–778

CDatabase class, 930–931, 983–984

CDBException class, 930–931

CDC class, 712–720

CEditView class, 660

CElement class, 736–737

cells, database tables

- alignment, 1100
- background color, 1099–1100
- buttons, 1095
- CellStyle property, 1109
- ColumnIndex property, 1109
- default styles, 1101–1108
- DesiredType property, 1109
- dynamic styles, 1108–1114
- fonts, 1099–1100
- foreground color, 1099–1100
- formatting, 1100
- FormattingApplied property, 1109
- header, customizing, 1101
- highlighting on mouseover, 1110–1114
- images, 1096
- non-header, customizing, 1101–1108
- padding, 1100
- properties, 1109
- RowIndex property, 1109
- selection color, 1099–1100
- Value property, 1109
- wrapping text, 1100

CellStyle property, 1109

CFieldExchange class, 930–931

CFormView class, 660

CFrameWnd class, 642

CHAR data type, 618

character arrays, 166–169

character data type, 55–56

check boxes, in cells, 1095

child windows, 615

CHtmlEditView class, 660

CHtmlView class, 660

circles, drawing

- Arc() function, 715–717
- CCircle class, 744–746
- Ellipse() function, 715–717
- with the mouse, 722–723, 744–746

class constructors

- adding to a class, 343–345
- copy, 356–358
- copy constructors, 356–358, 405–407, 490–495
- default, 345–347
- default parameter values, assigning, 347–349

defining, example, 447–448
 definition, 343
 derived classes, 482–486
 initialization lists, 350
 no-arg, 345–347
 reference class, 549–551

class destructors

calling the wrong one, 511–516
 default, 400
 definition, 399–400
 dynamic memory allocation, 402–405
 example, 400–402
 virtual, 511–516

class interface, 436**class libraries, creating, 532–535****class objects. See also class constructors; class destructors.**

arrays of, 363–364
 declaring, 336–337
 definition, 335
 pointers to, 368–371
 protecting, 360–361
 references to, 371–372

class templates

creating objects from, 431–433
 defining, 428–431
 definition, 427
 instantiating, 427–428
 with multiple parameters, 434–436
 template member functions, 430–431

Class View, 10, 17**classes. See also CBox class; class constructors; class destructors; collection classes; inheritance.**

:: (colons), scope resolution operator, 341–342
 abstract, 505–508
 access control, 336, 479–482, 489–490
 application, 653
 base
 accessing private members, 480–482
 definition, 475–476
 deriving classes from, 476–479
 direct, 475–476
 indirect, 475–476, 508–511
 pointers to, 501–503
 CArray template, 761–763
 CBox
 analyzing objects, 441–445
 class implementation, 444

 combining objects, 439–441
 comparing objects, 438–439
 defining
 the constructor, 447–448
 data members, 446–447
 function members, 448–453
 global functions, 453–455
 source code, 437–438
 using, 455–458

CCircle, 744–746

CCurve

 defining, 775–777
 drawing curves, 746
 exercising, 778–779
 implementing, 777–778

CDatabase, 930–931, 983–984

CDBException, 930–931

CDC, 712–720

CEditView, 660

CElement, 736–737

CFieldExchange, 930–931

CFormView, 660

CFrameWnd, 642

CHtmlEditView, 660

CHtmlView, 660

CLine, 737–741

CList template, 774–775. *See also* curves, drawing.
 arguments, 773–774
 CCurve class, 775–779
 description, 763–768

CListView, 660

CMap template, 768–771

CMyApp, 653–654

CMyDoc, 653–654

CMyView, 653–654

CMyWnd, 653–654

copying, 356–358

CPrintInfo, 886

CRecordset, 930–931

CRecordView, 930–931

CRectangle, 742–743

CRichEditView, 660

CScrollView, 660

CString, 858–859

CText, 861–862

CTreeView, 660

CTypedPtrList, 771

CView, 660

classes (continued)

classes (continued)

- CWinApp, 641–642
 - data members
 - accessing, 337–339
 - definition, 335–336
 - private, 350–353
 - static, 365–367
 - Debug, 602–611
 - defining, 336
 - definition, 335
 - derived. *See also* inheritance.
 - from base classes, 476–479
 - constructors, 482–486
 - copy constructors, 490–495
 - creating, 476–479
 - definition, 475
 - pointers to, 501–503
 - reference classes, C++/CLI, 524–526
 - value classes, C++/CLI, 520–526
 - document, 653
 - encapsulation, 335
 - fields, 336
 - frame window, 653
 - friend, 495–496
 - friend functions, 354–356
 - generic
 - collection classes, 555–561
 - defining, 551–552
 - Dictionary<TKey, TValue>, generic dictionary, 558
 - interface classes, 554–555
 - LinkedList<T>, generic doubly linked list, 557
 - List<T>, generic list, 556–557
 - using, 552–561
 - history of, 334
 - incomplete definitions, 504
 - inline functions, 342–343
 - instances
 - counting, 366–367
 - definition, 335
 - instantiation, 335
 - interface, 526–531
 - member functions
 - adding to a class, 339–341
 - defined outside the class, 362–363
 - definition, 335–336
 - positioning, 341–342
 - protecting, 361–362
 - static, 367–368
 - members, 335–336
 - nesting, 516–519
 - OOP (object-oriented programming), 335
 - operations, 334–335
 - overview, 332–334
 - private members, 350–353
 - properties
 - default indexed, 388
 - definition, 381
 - indexed, 381, 388–393
 - named indexed, 388
 - read-only, 381
 - reserved names, 394
 - scalar, 381, 382–388
 - static, 393–394
 - trivial scalar, 384–388
 - write-only, 381
 - protected class members, 486–489
 - serialization, 874
 - System::Data::DataColumn, 1090
 - System::Data::DataRow, 1090
 - System::Data::DataSet, 1090
 - System::Data::DataTable, 1090
 - this pointer, 358–360
 - Trace, 602–611
 - unions, 410
 - view, 653, 711–712, 954–957
 - virtual destructors, 511–516
 - virtual functions
 - definition, 499
 - description, 497–501
 - pure, 505
 - references with, 503–504
- classes, C++/CLI**
- class properties
 - default indexed, 388
 - definition, 381
 - indexed, 381, 388–393
 - named indexed, 388
 - read-only, 381
 - reserved names, 394
 - scalar, 381, 382–388
 - static, 393–394
 - trivial scalar, 384–388
 - write-only, 381
 - initonly fields, 394–395
 - literal fields, 377–378. *See also* initonly fields.
 - operator overloading
 - (minus signs), decrement operator, 467
 - ++ (plus signs), increment operator, 467

- in reference classes, 467–470
- in value classes, 461–466
- reference class types, defining, 378–381
- static constructors, 396
- `ToString()` function, 376–377
- value classes
 - defining, 373–378
 - deriving, 520–526
 - overloading, 461–466
- `Clear()` **function, 1094**
- CLI (Common Language Infrastructure), 2**
- client area**
 - definition, 615
 - drawing, 634–636
 - redrawing, 731–732
- client coordinates, 789–792**
- CLine class, 737–741**
- CList template class, 763–768. See also curves, drawing.**
 - arguments, 773–774
 - `CCurve` class
 - defining, 775–777
 - exercising, 778–779
 - implementing, 777–778
 - drawing curves, 774–775
- CListView class, 660**
- CLR (Common Language Runtime). See also C++/CLI programs.**
 - console applications
 - description, 6
 - sample project, 24–27
 - description, 2–3
- CMap template class, 768–771**
- CMyApp class, 653–654**
- CMyDoc class, 653–654**
- CMyView class, 653–654**
- CMyWnd class, 653–654**
- CObject-based classes, serialization, 872**
- collection classes. See also classes.**
 - arrays
 - `CArray` template class, 761–763
 - collections of objects, 761–763
 - definition, 760
 - collections of objects
 - arrays, 761–763
 - `CArray` template class, 761–763
 - `CList` template class, 763–768
 - `CMap` template class, 768–771
 - helper functions, 761–763
 - lists, 763–768
 - definition, 759–760
 - generic, 555–561
 - lists
 - `AddHead()` function, 764–766
 - adding elements, 764–766
 - `AddTail()` function, 764–766
 - `CList` template class, 763–768
 - `CompareElements()` function, 767
 - `ConstructElements()` function, 768
 - counting elements, 773
 - definition, 760
 - deleting elements, 767–768
 - `DestructElements()` function, 768
 - empty, verifying, 773
 - `Find()` function, 766–767
 - `GetHeadPosition()` function, 766
 - `GetTailPosition()` function, 766
 - helper functions, 768
 - `InsertAfter()` function, 765–766
 - `InsertBefore()` function, 765–766
 - iterating through, 766
 - pointers to elements, 772–773
 - `RemoveHead()` function, 767–768
 - `RemoveTail()` function, 767–768
 - removing elements, 772–773
 - searching, 766–767
 - searching for pointers, 773
 - maps
 - `CMap` template class, 768–771
 - definition, 760
 - hashing addresses, 768–769
 - `HashKey()` function, 770–771
 - helper functions, 770
 - `Lookup()` function, 769
 - retrieving objects from, 769
 - storing objects in, 769
 - shapes, 760
 - typed pointers
 - `AddHead()` function, 772
 - `AddTail()` function, 772
 - `CTypedPtrList` class, 771
 - `CTypedPtrList` operators, 771–773
 - `Find()` function, 773
 - `FindIndex()` function, 773
 - `GetAt()` function, 772
 - `GetCount()` function, 773
 - `GetHead()` function, 772
 - `GetHeadPos()` function, 773
 - `GetNext()` function, 772
 - `GetPrev()` function, 772

collection classes (continued)

typed pointers

- GetTail() function, 772
- GetTailPos() function, 773
- InsertAfter() function, 773
- InsertBefore() function, 773
- IsEmpty() function, 773
- RemoveAll() function, 772
- RemoveAt() function, 773
- RemoveHead() function, 772
- RemoveTail() function, 772
- SetAt() function, 773

types of collection, 760

type-safe handling, 760–761

collections of objects

arrays, 761–763

CArray template class, 761–763

CList template class, 763–768

CMap template class, 768–771

helper functions, 761–763

lists, 763–768

colons (::), scope resolution operator

classes, 341–342

definition, 26

example, 93–94

color

BackColor property, 1099–1100

background, cells, 1099–1100

drawing in, 717–721

ForeColor property, 1099–1100

foreground, cells, 1099–1100

pen, 747–748

selected cells, 1099–1100

SelectionBackColor property, 1099–1100

SelectionForeColor property, 1099–1100

column types, 1095–1099

ColumnIndex property, 1109

columns, database tables

adding, 1096

binding to controls, 1123–1127

buttons in cells, 1095

check boxes, 1095

defining, 1093–1099

drop-down lists, 1096

editing, 1096–1098

images in cells, 1096

links, displaying, 1096

width, setting, 1098–1099

combo boxes, 819, 1068–1070, 1072

ComboBox control, 1068–1070, 1072

comma (,), expression separator, 76–77

command line output, 63–64, 104

command line pointer, 622

COMMAND messages, 689

command messages, 682–683

command update handler, 697–699

comments, 44–45

CommitTrans() function, 983–984

common controls, 819

Common Language Infrastructure (CLI), 2

Common Language Runtime (CLR). See CLR (Common Language Runtime).

Common Type System (CTS), 3

CompareElements() function, 767

comparing values. See also loops.

&& (ampersands), logical AND, 125

! (exclamation mark), logical NOT, 126

|| (vertical bars), logical OR, 125–126

case labels, 130

C++/CLI, 150–156

conditional operator, 127–129

if statement

description, 117–118

extended, 120–122

nesting, 118–120

if-else statement, nesting, 122–124

logical operators, 124–127

loops

for, 133–139

continue statement, 139–140

definition, 132–133

do-while, 146–147

for each, 153–156

floating-point counters, 143

infinite, 137–139

multiple counters, 136–137

nesting, 147–150

non-integer counters, 140–143

while, 143–145

member access operators, 153

relational operators, 115–116

switch statement, 129–131

unconditional branching, 132

compiler, IDE, 9

compound statements, 49–50

conditional operator, 127–129

configurations, IDE, 19

console applications

- automatic generation, 50–51
- CLR
 - description, 6
 - sample projects, 24–27, 101–104
- description, 6
- sample project, 21–23
- Win32
 - description, 6
 - sample project, 13–21

const modifier

- description, 70–71
- function parameters, 249–250
- member functions, 361–362
- member functions defined outside the class, 362–363
- objects, 360–361

constant pointers, 183–185

constants. See *initially fields*; *literal fields*.

constants, pointers to, 183–185

`ConstructElements()` **function, 768**

constructors. See *class constructors*.

context menus

- associating with classes, 795–796
- checking items, 800–802
- choosing, 797–802
- creating, 794–795
- deleting elements, 807–808
- drawing highlighted elements, 806–807
- highlighting elements, 802–806
- identifying selected elements, 798–800
- masked elements, 814–815
- moving elements, 808–814
- positioning elements, 813–814
- servicing menu messages, 807–814
- Windows Forms
 - creating, 1049
 - responding to, 1079–1086
- `WM_MOUSEMOVE` handler, 810–811

`ContextMenuStrip` **control, 1049**

Context-sensitive help **option, 659**

continue **statement, 139–140**

`ContinueRouting()` **method, 698**

control menu, 616

control notification messages, 681

`ControlBox` **property, 1068**

controls. See *also dialog boxes*.

- adding to dialog boxes, 820–822
- appearance, 1102–1103

buttons

- in cells, 1095
- clicks, handling, 1076–1079, 1083–1086
- definition, 819
- illustration, 819
- labels, changing, 992–993
- movement, detecting, 725
- categories of, 819
- column format, 1106–1108
- column headers, 1105–1106
- combo boxes, 819
- common, 819
- database
 - appearance, 1102–1103
 - binding to columns, 1123–1127
 - binding to data sources, 1115–1120
 - column format, 1106–1108
 - column headers, 1105–1106
 - setting up, 1103–1104
- database update
 - button label, changing, 992–993
 - Cancel button, visibility, 993–994
 - canceling the update, 997–998
 - description, 990–991
 - edit controls, enabling/disabling, 991–992
 - expediting the update, 996–997
 - Record menu, disabling, 994–996
- edit boxes
 - creating, 856–857
 - creating text elements, 863–865
 - creating the dialog classes, 858–859
 - `CString` class, 858–859
 - `CText` class, 861–862
 - definition, 819
 - drawing `CText` objects, 862
 - illustration, 819
 - moving `CText` objects, 862
 - text elements, defining, 860–862
 - Text menu item, 859–860
- group boxes, 821–822
- horizontal guides, 821
- illustration, 819
- linking to recordsets, 946–948
- list boxes
 - creating, 853–855
 - creating the dialog class, 854–855
 - definition, 819
 - displaying, 855
 - illustration, 819
 - scaling, 852–853

controls (continued)

controls (continued)

- positioning, 822
- scrollbars, 819
- setting up, 1103–1104
- spin buttons
 - creating, 835–837
 - dialog data exchange, 840
 - displaying, 841–842
 - initializing the dialog, 840–841
 - scale dialog class, 838–841
 - scale menu item, 835
 - scale toolbar button, 835
 - scaling, 835
 - validation, 840
- static, 819
- tab sequence, 838
- testing, 822

cooperative multitasking, 629–630

coordinate systems

- client coordinates, 789–792, 843
- device coordinates, 843
- drawing, 708
- logical coordinates, 789–790
- page coordinates, 843
- screen coordinates, 843

copy class constructors, 356–358

copy constructors, 405–407, 490–495

copying

- classes, 356–358
- rows, 1094

CPrintInfo **class**, 886

CreateElement() **function**, 751–753

CRecordset **class**, 930–931

CRecordset **operations**, 980–982

CRecordView **class**, 930–931

CRectangle **class**, 742–743

CRichEditView **class**, 660

_CRTDBG... **control bits**, 595–596

_crtDbgFlag **flag**, 595

crtdbg.h **header**, 594–595

CrtSetReportFile() **function**, 596–597

CrtSetReportMode() **function**, 596–597

CScrollView **class**, 660

CString **class**, 858–859

CText **class**, 861–862

CTreeView **class**, 660

Ctrl key detection, 728

CTS (Common Type System), 3

CTypedPtrList **class**, 771

CTypedPtrList **operators**, 771–773

curly braces ({ }), **function syntax**, 46

current position, 713

CurrentPosition() **function**, 1121

cursor position, recording, 729–731

curves, drawing

CCurve class

- defining, 775–777

- drawing curves, 746

- exercising, 778–779

- implementing, 777–778

CList template class, 774–775

- example, 723–724

CView **class**, 660

CWinApp **class**, 641–642

D

dash-dot lines, 718

dashed lines, 718

data, displaying. *See* DataGridView control.

data members

- accessing, 337–339

- defining, example, 446–447

- definition, 335–336

- private, 350–353

- static, 365–367

data sources. *See* databases.

data types. *See also* variables.

- fundamental. *See also* Boolean variables.

- bool, 57

- Boolean variables, 57

- boxing, 101

- C++/CLI, 99–104, 109–111

- char, 55–56

- character, 55–56

- const modifier, 70–71

- double, 58

- enumeration constants, 62, 109–111

- float, 58

- floating-point, 57–58

- int, 54

- integer type modifiers, 56–57

- integer variables, 54–55

- literals, 59

- long, 55

- long double, 58
- minimum/maximum values, 56–57
- short, 54
- signed modifier, 56–57
- summary of, 58–59
- synonyms for, 60
- true/false, 57
- typedef keyword, 60
- unboxing, 101
- wchar_t, 56
- wide character, 56
- Windows programs
 - BOOLEAN, 618
 - BYTE, 618
 - CHAR, 618
 - CTS (Common Type System), 3
 - DWORD, 618
 - HANDLE, 618
 - HBRUSH, 618
 - HCURSOR, 618
 - HDC, 618
 - HINSTANCE, 618
 - LPARAM, 619
 - LPCSTR, 619
 - LPHANDLE, 619
 - LRESULT, 619
 - overview, 332–334
 - WORD, 619
- database applications, creating**
 - dynaset recordsets, 935–936
 - generating an MFC ODBC program, 933–936
 - program structure
 - database/recordset data transfer, 940–941
 - DDX_FieldText() function, 946–948
 - DoFieldExchange() function, 940–941
 - linking controls to recordsets, 946–948
 - querying the database, 938–940
 - record view, 941–943
 - recordsets, 937–941
 - RFX_() function, 941
 - view dialog, creating, 943–946
 - registering an ODBC database, 931–933
 - snapshot recordsets, 935–936
- database applications, Windows Forms**
 - accessing data, 1090–1091
 - AddNew() function, 1121
 - BindingNavigator control, 1120–1123
 - BindingSource component, 1115–1120
 - bound mode, 1114–1115
 - cells
 - alignment, 1100
 - background color, 1099–1100
 - buttons, 1095
 - CellStyle property, 1109
 - ColumnIndex property, 1109
 - default styles, 1101–1108
 - DesiredType property, 1109
 - dynamic styles, 1108–1114
 - fonts, 1099–1100
 - foreground color, 1099–1100
 - formatting, 1100
 - FormattingApplied property, 1109
 - header, customizing, 1101
 - highlighting on mouseover, 1110–1114
 - images, 1096
 - non-header, customizing, 1101–1108
 - padding, 1100
 - properties, 1109
 - RowIndex property, 1109
 - selection color, 1099–1100
 - Value property, 1109
 - wrapping text, 1100
 - columns
 - adding, 1096
 - binding to controls, 1123–1127
 - buttons in cells, 1095
 - check boxes, 1095
 - defining, 1093–1099
 - drop-down lists, 1096
 - editing, 1096–1098
 - images in cells, 1096
 - links, displaying, 1096
 - width, setting, 1098–1099
 - controls
 - appearance, 1102–1103
 - binding to columns, 1123–1127
 - binding to data sources, 1115–1120
 - column format, 1106–1108
 - column headers, 1105–1106
 - setting up, 1103–1104
 - CurrentPosition() function, 1121
 - data source classes, 1090
 - DataGridView control
 - Add() function, 1094
 - AddCopy() function, 1094
 - bound mode, 1093
 - Clear() function, 1094
 - column types, 1095–1099

database applications, Windows Forms (continued)

- DataGridView control
 - DataGridViewButtonColumn column, 1095
 - DataGridViewCheckBoxColumn column, 1095
 - DataGridViewComboBoxColumn column, 1096
 - DataGridViewImageColumn column, 1096
 - DataGridViewLinkColumn column, 1096
 - definition, 1090
 - displaying data, 1091–1093
 - Insert() function, 1094
 - InsertCopy() function, 1094
 - modes, 1092–1099
 - Remove() function, 1094
 - RemoveAt() function, 1094
 - unbound mode, 1093–1099
 - virtual mode, 1093
- DataGridView control, customizing
 - Alignment property, 1100
- databases. See also DataGridView control.**
 - customer details
 - dialog resource, 968
 - filter parameters, 972–973
 - filters, 970–972
 - linking dialogs, 973–975
 - recordset, 967
 - view class, 968–970
 - fields, 922
 - keys, 923
 - MFC support for
 - CDatabase class, 930–931
 - CDBException class, 930–931
 - CFieldExchange class, 930–931
 - classes, 930–931
 - CRecordset class, 930–931
 - CRecordView class, 930–931
 - ODBC approach, 929–930
 - OLE DB approach, 929–930
 - m_strSort() function, 948
 - overview, 922–924
 - primary keys, 923
 - RDBMS (relational database management systems), 923
 - records, 922
 - recordsets
 - description, 937–941
 - dynaset, 935–936
 - snapshot, 935–936
 - sorting, 948–950

- recordsets, second object
 - customizing, 957–961
 - dialog resources, creating, 954–955
 - filter parameters, defining, 958–960
 - filters, adding, 958
 - initializing record view, 960–961
 - m_strFilter() function, 958
 - multiple table views, 961–966
 - OnActiveView() function, 965–966
 - recordset class, adding, 950–954
 - switching views, 961–965
 - view activation, 965–966
 - view class, adding, 954–957
 - viewing product orders, 966–967
- relational, 921
- SQL
 - choosing records, 925–926
 - joining tables, 926–928
 - retrieving data, 924–926
 - sorting records, 929
- tables, 922
- window caption, modifying, 949–950
- databases, updating**
 - adding records, 980–982
 - adding rows to tables
 - adding orders, 1023–1028
 - dialog switching, 1010–1014
 - order data, storing, 1019–1021
 - order entry process, 1000–1001
 - order IDs, creating, 1014–1019
 - overview, 999–1000
 - recordsets, creating, 1002
 - recordsets views, creating, 1002–1006
 - resources, adding controls, 1006–1010
 - resources, creating, 1001
 - selecting products, 1021–1023
 - setting dates, 1020–1021
- AddNew() function, 980–982
- BeginTrans() function, 983–984
- canceling updates, 980–982, 997–998
- CancelUpdate() function, 980–982
- CDatabase class, 983–984
- CommitTrans() function, 983–984
- completing an update, 980–982
- controls
 - button label, changing, 992–993
 - Cancel button, visibility, 993–994
 - canceling the update, 997–998

- description, 990–991
- edit controls, enabling/disabling, 991–992
- expediting the update, 996–997
- Record menu, disabling, 994–996
- CRecordset operations, 980–982
- Delete() function, 980–982
- deleting records, 980–982
- Edit() function, 980–982
- editing records, 980–982
- example, 984–988
- inhibiting update, 988–990
- optimistic record locking, 982
- pessimistic record locking, 982
- record locking, 982
- Rollback() function, 983–984
- rolling back an update, 982–984
- starting an update, 988–990
- tables, adding rows to
 - adding orders, 1023–1028
 - dialog switching, 1010–1014
 - order data, storing, 1019–1021
 - order entry process, 1000–1001
 - order IDs, creating, 1014–1019
 - overview, 999–1000
 - recordsets, creating, 1002
 - recordsets views, creating, 1002–1006
 - resources, adding controls, 1006–1010
 - resources, creating, 1001
 - selecting products, 1021–1023
 - setting dates, 1020–1021
- transactions, 982–984
- Update() function, 980–982
- update mode
 - button label, changing, 992–993
 - Cancel button, visibility, 993–994
 - canceled the update, 997–998
 - description, 990–991
 - edit controls, enabling/disabling, 991–992
 - expediting the update, 996–997
 - Record menu, disabling, 994–996
- validating operations, 981
- DataGridView **control**
 - Add() function, 1094
 - AddCopy() function, 1094
 - bound mode, 1093
 - Clear() function, 1094
 - column types, 1095–1099
 - customizing
 - Alignment property, 1100
 - AlternatingRowsDefaultCellStyle property, 1101
 - BackColor property, 1099–1100
 - DataGridViewCellStyle object, 1099–1100
 - DefaultCellStyle property, 1101
 - Font property, 1099–1100
 - ForeColor property, 1099–1100
 - Format property, 1100
 - InheritedStyle property, 1100–1101
 - Padding property, 1100
 - RowsDefaultCellStyle property, 1101
 - SelectionBackColor property, 1099–1100
 - SelectionForeColor property, 1099–1100
 - Style property, 1101
 - System::Collections::IList interface, 1114
 - System::ComponentModel::BindingList interface, 1114
 - System::ComponentModel::BindingListView interface, 1114
 - System::ComponentModel::IListSource interface, 1114
 - WrapMode property, 1100
 - DataGridViewButtonColumn column, 1095
 - DataGridViewCheckBoxColumn column, 1095
 - DataGridViewComboBoxColumn column, 1096
 - DataGridViewImageColumn column, 1096
 - DataGridViewLinkColumn column, 1096
 - definition, 1091
 - displaying data, 1091–1093
 - Insert() function, 1094
 - InsertCopy() function, 1094
 - modes, 1092–1099
 - Remove() function, 1094
 - RemoveAt() function, 1094
 - unbound mode, 1093–1099
 - virtual mode, 1093
- DataGridViewButtonColumn **column, 1095**
- DataGridViewCellStyle **object, 1099–1100**
- DataGridViewCheckBoxColumn **column, 1095**
- DataGridViewComboBoxColumn **column, 1096**
- DataGridViewImageColumn **column, 1096**
- DataGridViewLinkColumn **column, 1096**
- DDX_FieldText() **function, 946–948**
- Debug **class, 602–611**
- Debug **configuration, 570**

debugger

- definition, 568
- starting, 570, 573–576
- windows, 575

debugging. See also bugs.

- adding code for, 578–585
- assert() function, 578–580
- assertions, 578–580
- basic operations, 568–570
- breakpoints, 570–572
- call stack, 585–587
- C++/CLI
 - Assert() function, 607–608
 - assertions, 607–608
 - controlling output, 605–607
 - Debug class, 602–611
 - generating output, 603–604
 - Indent() function, 604–605
 - indenting output, 604–605
 - listeners, 604
 - output destination, 604
 - Trace class, 602–611
 - trace switches, 605–607
 - Unindent() function, 604–605
 - Write() function, 603–604
 - WriteIf() function, 603–604
 - WriteLine() function, 603–604
 - WriteLineIf() function, 603–604
- changing variables, 577–578
- _CRTDBG... control bits, 595–596
- _crtDbgFlag flag, 595
- crtDBG.h header, 594–595
- CrtSetReportFile() function, 596–597
- CrtSetReportMode() function, 596–597
- definition, 565
- description, 565–566
- dynamic memory, 593–601
- finding the next bug, 593
- free store operations, 594–601
- in IDE, 19–23
- #ifdef/#endif directives, 580–581
- inspecting variables, 576–577
- memory leaks, 594–601
- NDEBUG symbol, 578–579
- null pointers, 576
- setting a watch, 576
- stepping through code, 587–590
- testing the extended class, 591–593
- tracepoints, 572–573

DECLARE_DYNAMIC() **macro, 872**

DECLARE_DYNCREATE() **macro, 869, 872**

DECLARE_SERIAL() **macro, 872**

declaring

- pointers, 173–174
- variables, 52–53, 91

_declspec **keyword, 916–917**

decoding windows messages, 633–636

decorated names, 912

decrementing variables, 74–76

default indexed properties, 388

DefaultCellStyle **property, 1101**

definition, 759–760

delegates

- calling, 540–541
- creating, 537–541
- declaring, 537
- definition, 536
- unbound, 541–545

Delete() **function, 980–982**

delete **operator, 193**

deleting

- context menu elements, 807–808
- database records, 980–982
- list elements, 767–768
- rows, 1094
- shapes from documents, 793–794

de-referencing pointers, 174

derived classes

- from base classes, 476–479
- class constructors, 482–486
- constructors, 482–486
- copy constructors, 490–495
- creating, 476–479
- definition, 475
- pointers to, 501–503
- reference classes, C++/CLI, 524–526
- value classes, C++/CLI, 520–526

DesiredType **property, 1109**

DestructElements() **function, 768**

device context, 634–636, 709, 893

Dialog based **option, 657–658**

dialog boxes

- adding, 820–822
- adding a class, 823–824
- closing, 827
- controls
 - adding, 820–822
 - enabling, 831
 - initializing, 828–830

- radio button messages, 830–831
- creating elements, 833–834
- description, 817–818
- displaying, 824–827
- enabling, 831
- illustration, 818
- initializing, 828–830
- linking, 973–975
- modal, 824
- modeless, 824
- pen widths, 832–833
- radio button messages, 830–831
- scale factor
 - document size, 844
 - mapping mode, 844–846
 - scaleable mapping modes, 842–844
 - scrolling, 846–848
- scrollbars, 847–848
- status bars
 - adding to frames, 848–850
 - overview, 848
 - parts of, 850–851
 - updating, 851–852
- testing, 822, 834–835
- Windows Forms
 - adding, 1068–1075
 - button clicks, handling, 1076–1079, 1083–1086
 - ComboBox control, 1068–1070, 1072
 - context menu, responding to, 1079–1086
 - ControlBox property, 1068
 - creating, 1056–1062
 - DialogResult property, 1057, 1063–1066
 - event handler, Reset menu, 1067–1068
 - FormBorderStyle property, 1068
 - getting data from, 1070–1073
 - Help > About menu, 1075–1076
 - input controls, disabling, 1073–1074
 - Limits menu, 1074–1075
 - MaximizeBox property, 1068
 - MinimizeBox property, 1068
 - NumericUpDown control, 1068–1070, 1073
 - Show() function, 1062–1066
 - ShowDialog() function, 1062–1063
 - Text property, 1068
 - using, 1062–1068
 - validating input, 1063–1066
- dialog button events, 1058–1060**
- dialog objects, creating, 1061–1062**
- DialogResult **property, 1057, 1063–1066**
- Dictionary<TKey, TValue>, **generic dictionary, 558**
- direct base classes, 475–476**
- display content, 634–636**
- displaying data. See DataGridView control.**
- displaying graphics, 713–717**
- dllexport **attribute, 916–917**
- DllMain() **function, 906, 910–911**
- DLLs (dynamic link libraries)**
 - AFX_EXT_CLASS keyword, 912–913, 916
 - contents
 - choosing, 907–908
 - description, 906
 - _declspec keyword, 916–917
 - decorated names, 912
 - dllexport attribute, 916–917
 - DllMain() function, 906
 - early binding, 904–906
 - exporting
 - symbols, 917–920
 - variables and functions, 916–917
 - functional description, 903–906
 - importing symbols, 917
 - interface, 906, 913
 - late binding, 904–906
 - load-time dynamic linking, 904–906
 - MFC extension, 906–907
 - ordinals, 912
 - overview, 901–903
 - regular, dynamically linked, 907
 - regular, statically linked, 907
 - run-time dynamic linking, 904–906
 - varieties, 906–907
 - writing extension DLLs
 - adding classes, 911–912
 - building DLLs, 914
 - DllMain() function, 910–911
 - example, 914–915
 - exporting classes, 912–913
 - overview, 908–910
 - required files, 915–916
- docking toolbars, 12**
- document class, 653**
- document interfaces, MFC, 650**
- documentation, IDE, 12–13**
- documents. See also printing documents; serialization; views.**
 - context menus

- associating with classes, 795–796
- checking items, 800–802
- choosing, 797–802
- creating, 794–795
- deleting elements, 807–808
- drawing highlighted elements, 806–807
- highlighting elements, 802–806
- identifying selected elements, 798–800
- masked elements, 814–815
- moving elements, 808–814
- positioning elements, 813–814
- servicing menu messages, 807–814
- WM_MOUSEMOVE handler, 810–811
- creating
 - adding elements, 783–784
 - CTypedPtrList template, 779–784
 - document destructor, 780–781
 - drawing the document, 781–783
 - OnDraw() function, 781–783
 - pointers to shape classes, 779–784
 - RectVisible() function, 783
- definition, 650
- deleting shapes, 793–794
- linking to views, 652–653
- moving shapes, 793–794
- moving text, 882–883
- pen widths, 832–833
- recording changes, 874–876
- size, getting, 889–890
- size, scale factor, 844
- template classes, 653
- templates, 652–653
- Document/View architecture support **option, 657–658**
- DoFieldExchange() **function, 940–941**
- dotted lines, 718**
- double **data type, 58**
- double quotes (“...”), special character indicator, 65**
- do-while **loops, 146–147**
- Draw() **function, 738–739**
- drawing in windows**
 - alternating dash-dot lines, 718
 - Arc() function, 716–717
 - brushes, 719–721
 - CDC class, 712–720

- circles
 - Arc() function, 715–717
 - CCircle class, 744–746
 - Ellipse() function, 715–717
 - with the mouse, 722–723, 744–746
- in color, 717–721
- coordinate system, 708
- current position, 713
- curves
 - CCurve class, 746, 775–778, 778–779
 - CList template class, 774–775
 - example, 723–724
- dashed lines, 718
- device context, 709
- displaying graphics, 713–717
- dotted lines, 718
- GDI (Graphical Device Interface), 709–711
- hatching, 719–720
 - HS_BDIAGONAL hatching, 720
 - HS_CROSS hatching, 720
 - HS_DIAGCROSS hatching, 720
 - HS_FDIAGONAL hatching, 720
 - HS_HORIZONTAL hatching, 720
 - HS_VERTICAL hatching, 720
- inside a frame, 718
- lines
 - alternating dash-dot, 718
 - CLine class, 737–741
 - dashed, 718
 - dotted, 718
 - LineTo() function, 714–715
 - with the mouse, 737–741
 - solid, 718
 - straight, 714–715
- LineTo() function, 714–715
- mapping modes, 709–711
 - MM_ANISOTROPIC mapping mode, 710–711
 - MM_HIENGLISH mapping mode, 709–711
 - MM_HIMETRIC mapping mode, 709–711
 - MM_ISOTROPIC mapping mode, 709–711
 - MM_LOENGLISH mapping mode, 709–711
 - MM_LOMETRIC mapping mode, 709–711
 - MM_TEXT mapping mode, 709–711
 - MM_TWIPS mapping mode, 710–711
- mouse
 - black-and-white, 747
 - bounding rectangles, 739–741
 - button movement, detection, 725

- CCircle class, 744–746
- CCurve class, 746
- CElement class, 736–737
- circles, 744–746
- CLine class, 737–741
- CreateElement() function, 751–753
- creating elements, 751–753
- CRectangle class, 742–743
- Ctrl key detection, 728
- cursor position, recording, 729–731
- curves, 746
- Draw() function, 738–739
- drawing mode, 747–749
- element classes, 732–736
- Ellipse() function, 745–746
- InvalidateRect() function, 731–732
- left button detection, 725, 728
- lines, 737–741
- message handlers, 727–729, 747–753
- messages, capturing, 755–756
- messages, types of, 725–727
- middle button detection, 728
- MK_CONTROL flag, 728–729
- MK_LBUTTON flag, 728–729
- MK_MBUTTON flag, 728–729
- MK_RBUTTON flag, 728–729
- MK_SHIFT flag, 728–729
- movement, handling, 749–751
- normalized rectangles, 741
- OnMouseMove() handler, 749–751
- pen color, 747–748
- R2_BLACK mode, 747–748
- R2_COPYPEN mode, 747–748
- R2_MASKNOTOPEN mode, 748
- R2_MASKPEN mode, 748
- R2_MASKPENNOT mode, 747–748
- R2_MERGENOTOPEN mode, 747–748
- R2_MERGEOPEN mode, 748
- R2_MERGEOPENNOT mode, 747–748
- R2_NOP mode, 747–748
- R2_NOT mode, 747–748
- R2_NOTCOPYPEN mode, 747–748
- R2_NOTMASKPEN mode, 748
- R2_NOTMERGEPEN mode, 748
- R2_NOTXORPEN mode, 748
- R2_WHITE mode, 747–748
- R2_XORPEN mode, 748
- Rectangle() function, 743
- rectangles, 739–741, 742–743
- redrawing the client area, 731–732
- ReleaseCapture() function, 755–756
- right button detection, 728
- rubberbanding, 747, 750
- SetCapture() function, 755–756
- SetROP2() function, 747–750
- Shift key detection, 728
- temporary element storage, 734–736
- update region, 731
- WM_LBUTTONDOWN message, 725–726
- WM_LBUTTONUP message, 725, 727, 753
- WM_MOUSEMOVE message, 725, 726–727
- OnDraw() function, 711–712
- pen objects, 717–719
- positioning the drawing, 708
- PS_DASH pen style, 718
- PS_DASHDOT pen style, 718
- PS_DASHDOTDOT pen style, 718
- PS_DOT pen style, 718
- PS_INSIDEFRAME pen style, 718
- PS_NULL pen style, 718
- PS_SOLID pen style, 718
- rectangles, 722, 724
- rectangles, with the mouse
 - bounding rectangles, 739–741
 - CRectangle class, 742–743
 - normalized rectangles, 741
 - Rectangle() function, 743
 - rectangles, 722, 724, 739–741, 742–743
- solid lines, 718
- view class, 711–712
- window client area, 708
- WM_PAINT message, 708, 711, 731, 754–755
- drawing mode, 747–749**
- drop-down lists, in cells, 1096**
- d w prefix, 619–620**
- DWORD data type, 618**
- dynamic link libraries (DLLs). See DLLs.**
- dynamic memory, debugging, 593–601**
- dynamic memory allocation. See also free store.**
 - allocating, 193
- C++/CLI
 - arrays of arrays, 213–216
 - CLR arrays, 200–205
 - freeing memory, 198–199
 - garbage collection, 198–199
 - handles, 199–200

dynamic memory allocation (continued)

C++/CLI

- jagged arrays, 213–216
- multidimensional arrays, 209–213
- searching one-dimensional arrays, 206–209
- sorting one-dimensional arrays, 205–206
- string handling, 216–225
- tracking handles, 199–200

class destructors, 402–405

delete operator, 193

free store

- definition, 192–193
- example, 194–196

freeing, 193

handling allocation errors, 284–285

multidimensional arrays, 196–197

new operator, 193, 284–285

one-dimensional arrays, 194–196

references. *See also* pointers; tracking references.

- declaring, 197–198
- definition, 197
- initializing, 197–198

`dynamic_cast` operator, 516

`dynaset` recordsets, 935–936

E

early binding, 904–906

edit boxes

creating, 856–857

`CString` class, 858–859

`CText` class, 861–862

definition, 819

dialog classes, creating, 858–859

drawing `CText` objects, 862

illustration, 819

moving `CText` objects, 862

text elements, creating, 863–865

text elements, defining, 860–862

`Text` menu item, 859–860

`Edit()` function, 980–982

editing

columns, 1096–1098

database records, 980–982

editor, IDE, 9

Editor tab, 10

element classes, 732–736, 877–879

elements, of arrays, 160

`Ellipse()` function, 745–746

`Enable()` method, 698

encapsulation, 335

`EndsWith()` function, 222–225

enumeration, 60–62

enumeration constants, 62, 109–111

equal sign (=), assignment operator, overloading, 418–422

equal signs (==), relational operator, 115–116

error handling, 23–24. *See also* exceptions.

escape sequences, 65–67

event handlers

- dialog button, 1058–1060
- menu items, 1049–1056
- `Reset` menu, 1067–1068

event-driven

- applications, 7
- Windows programs, 617

events

- creating, 545–546
- definition, 536
- handling, 547–548
- Windows applications, 7

exceptions. *See also* error handling.

- catch blocks, 281–282

- catching

 - C++/CLI, 308–309
 - example, 280–281

- C++/CLI, 308–309

- definition, 279–280

- in the MFC, 283

- throwing

 - C++/CLI, 308–309

 - description, 281

 - example, 280–281

- try blocks, 281–283

exclamation mark, equal sign (!=), relational operator, 115–116

exclamation mark (!), logical NOT, 126

.exe file extension, 19

executing programs, IDE, 20–21

explicit casts, 80–81

exporting

- classes from extension DLLs, 912–913
- symbols from DLLs, 917–920
- variables and functions from DLLs, 916–917

expressions

- casting
 - in assignment statements, 79–80
 - definition, 78
 - explicit casts, 80–81
 - old-style casts, 81
 - rules for, 78–79
- evaluating, 295–298, 316–317
- mixed type, casting, 80–81
- separating, 76–77

extension DLLs

- adding classes, 911–912
- building DLLs, 914
- DllMain() function, 910–911
- example, 914–915
- exporting classes, 912–913
- overview, 908–910
- required files, 915–916

F**fields**

- classes, 336
- databases, 922

file scope. See **global scope.**

filter parameters, databases, 958–960, 972–973

filters, databases, 958, 970–972

finalizers, 549–551

Find() **function, 766–767, 773**

FindIndex() **function, 773**

FixedDialog **property, 1056–1062**

float **data type, 58**

floating-point data types, 57–58

floating-point loop counters, 143

fn **prefix, 619–620**

Font **property, 1099–1100**

fonts, cells, 1099–1100

for each **loops, 153–156**

for **loops, 133–139**

ForeColor **property, 1099–1100**

foreground color, cells, 1099–1100

Format **property, 1100**

formatting cells, 1100

formatting output, 64–65, 104–107

FormattingApplied **property, 1109**

FormBorderStyle **property, 1056–1062, 1068**

forms. See **Windows Forms.**

frame window class, 653

frame windows, 642

free store

- debugging, 594–601
- definition, 192–193
- example, 194–196

freeing memory, 193, 198–199. See also **garbage collection.**

friend classes, 495–496

friend functions, in classes, 354–356

function members. See **member functions.**

function pointers

- as arguments, 275–277
- arrays of as arguments, 277
- declaring, 272–273
- definition, 271–272
- example, 273–275

functions

{ } (curly braces), function syntax, 46

Add(), 1094

AddCopy(), 1094

AddHead(), 764–766, 772

AddNew(), 980–982, 1121

AddTail(), 764–766, 772

Arc(), 716–717

arguments

arrays as, 243–247

arrays of function pointers as, 277

definition, 232

function pointers as, 275–277

passing to main(), 250–252, 267–268

passing-by-pointer, 241–243

passing-by-reference, 247–249

passing-by-value, 240–241

pointer notation, 244–245

variable number of, 252–254, 266–267

Assert(), 578–580, 607–608

BeginTrans(), 983–984

body, 46, 234–235

calling, 235–239

calling, recursively, 262–265

calls, counting, 260

CancelUpdate(), 980–982

C++/CLI, 265–268, 309–315

Clear(), 1094

CommitTrans(), 983–984

CompareElements(), 767

ConstructElements(), 768

functions (continued)

functions (continued)

CreateElement(), 751–753
CrtSetReportFile(), 596–597
CrtSetReportMode(), 596–597
CurrentPosition(), 1121
DDX_FieldText(), 946–948
Delete(), 980–982
description, 38
DestructElements(), 768
DllMain(), 906, 910–911
DoFieldExchange(), 940–941
Draw(), 738–739
Edit(), 980–982
Ellipse(), 745–746
EndsWith(), 222–225
executable statements, 46
Find(), 766–767, 773
FindIndex(), 773
generic, 309–315
GetAt(), 772
GetCount(), 773
GetFromPage(), 887
GetHead(), 772
GetHeadPos(), 773
GetHeadPosition(), 766
GetMaxPage(), 887
GetMinPage(), 887
GetNext(), 772
GetPrev(), 772
GetTail(), 772
GetTailPos(), 773
GetTailPosition(), 766
GetToPage(), 887
global variables, 260
HashKey(), 770–771
header, 233–234
Indent(), 604–605
inline, in classes, 342–343
Insert(), 1094
InsertAfter(), 765–766, 773
InsertBefore(), 765–766, 773
InsertCopy(), 1094
InvalidateRect(), 731–732
IsEmpty(), 773
LineTo(), 714–715
LookUp(), 769
main(), 46, 250–252, 267–268
MoveFirst(), 1121
MoveLast(), 1121
MoveNext(), 1121
MovePrevious(), 1121
m_strFilter(), 958
new, 536
OnActiveView(), 965–966
OnDraw(), 711–712
OnUpdate(), 785–787
overloading, 285–288
parameters
description, 232
initializing, 277–279
protecting, 249–250
prototypes, 235–239
Rectangle(), 743
recursive, 262–265
RegisterClassEx(), 625–627
ReleaseCapture(), 755–756
Remove(), 1094
RemoveAll(), 772
RemoveAt(), 773, 1094
RemoveCurrent(), 1121
RemoveHead(), 767–768, 772
RemoveTail(), 767–768, 772
return statement, 235
returning values from
addresses, 256–257
bad pointers, 255–256
pointers, 254–257
references, 258–260
terms, 298–299, 318
RFX_(), 941
Rollback(), 983–984
Serialize()
definition, 870
serializing documents, 876–877
shape classes, 879–881
SetAt(), 773
SetCapture(), 755–756
SetMaxPage(), 887
SetMinPage(), 887
SetROP2(), 747–750
Show(), 1062–1066
ShowDialog(), 1062–1063
StartsWith(), 222–225
static variables, 261–262
structure of, 233–235
templates, 288–291
ToString(), 376–377
TotalNumberOfItems(), 1121

trim(), 220–221
 Unindent(), 604–605
 Update(), 980–982
 UpdateAllViews(), 785–787
 uses for, 233
 using, 235–239
 WINAPI(), 622
 WindowProc(), 632–637
 WinMain()
 arguments, 622
 example, 631–632
 hInstance argument, 622
 hPrevInstance argument, 622
 lpCmdLine argument, 622
 nCmdShow argument, 622
 prototype, 621
 RegisterClassEx() function, 625–627
 WINAPI() function, 622
 WNDCLASSEX struct, 623–625
 Write(), 103–104, 603–604
 WriteIf(), 603–604
 WriteLine(), 103–104, 603–604
 WriteLineIf(), 603–604
functions, example (calculator)
 analyzing a number, 299–302, 318–319
 C++/CLI, 315–320
 expressions, evaluating, 295–298, 316–317
 extending the program, 304–305
 project description, 291–294
 returning values from terms, 298–299, 318
 running the program, 307–308
 source code, 302–303
 strings
 extracting substrings, 305–307, 319–320
 removing blanks, 294–295, 316
fundamental data types. See also Boolean variables;
 data types.
 bool, 57
 Boolean variables, 57
 boxing, 101
 C++/CLI, 99–104, 109–111
 char, 55–56
 character, 55–56
 const modifier, 70–71
 double, 58
 enumeration constants, 62, 109–111
 float, 58
 floating-point, 57–58
 int, 54

 integer type modifiers, 56–57
 integer variables, 54–55
 literals, 59
 long, 55
 long double, 58
 minimum/maximum values, 56–57
 short, 54
 signed modifier, 56–57
 summary of, 58–59
 synonyms for, 60
 true/false, 57
 typedef keyword, 60
 unboxing, 101
 wchar_t, 56
 wide character, 56

G

garbage collection, 198–199

GDI (Graphical Device Interface), 709–711

generic classes

 collection classes, 555–561

 defining, 551–552

 Dictionary<TKey, TValue>, generic dictionary,
 558

 interface classes, 554–555

 LinkedList<T>, generic doubly linked list, 557

 List<T>, generic list, 556–557

 using, 552–561

generic functions, 309–315

 GetAt() **function, 772**

 GetCount() **function, 773**

 GetFromPage() **function, 887**

 GetHead() **function, 772**

 GetHeadPos() **function, 773**

 GetHeadPosition() **function, 766**

 GetMaxPage() **function, 887**

 GetMinPage() **function, 887**

 GetNext() **function, 772**

 GetPrev() **function, 772**

 GetTail() **function, 772**

 GetTailPos() **function, 773**

 GetTailPosition() **function, 766**

 GetToPage() **function, 887**

global functions, defining, 453–455

global namespace scope. See global scope.

global scope, 91–94

global variables, 260

Graphical Device Interface (GDI)

Graphical Device Interface (GDI), 709–711

Graphical User Interface (GUI). See GUI (Graphical User Interface).

graphics, displaying, 713–717

greater than, equal sign (>=), relational operator, 115–116

greater than (>)

operator overloading, 415–417

OR, 84–85

relational operator, 115–116

group boxes, 821–822, 1042–1044

GroupBox control, 1042–1044

grouping controls, 821–822, 1042–1044

GUI (Graphical User Interface), Windows Forms

Button control, 1044–1046

context menus, 1049

ContextMenuStrip control, 1049

dialog boxes

adding, 1068–1075

button clicks, handling, 1076–1079, 1083–1086

ComboBox control, 1068–1070, 1072

context menu, responding to, 1079–1086

ControlBox property, 1068

creating, 1056–1062

DialogResult property, 1057, 1063–1066

event handler, Reset menu, 1067–1068

FormBorderStyle property, 1068

getting data from, 1070–1073

Help > About menu, 1075–1076

input controls, disabling, 1073–1074

Limits menu, 1074–1075

MaximizeBox property, 1068

MinimizeBox property, 1068

NumericUpDown control, 1068–1070, 1073

Show() function, 1062–1066

ShowDialog() function, 1062–1063

Text property, 1068

using, 1062–1068

validating input, 1063–1066

dialog button events, 1058–1060

dialog objects, creating, 1061–1062

event handler, dialog button, 1058–1060

event handlers, menu items, 1049–1056

FixedDialog property, 1056–1062

FormBorderStyle property, 1056–1062

GroupBox control, 1042–1044

grouping controls, 1042–1044

list boxes, 1058–1061

menu items, event handlers, 1049–1056

menus, 1037–1038

overview, 1035–1036

submenus, 1038–1040

tab controls, 1040–1042

TabControl control, 1040–1042

testing, 1048–1049

Web browser controls, 1047–1048

WebBrowser control, 1047–1048

H

h prefix, 619–620

HANDLE data type, 618

handles, 199–200, 622

hashing addresses, 768–769

HashKey() function, 770–771

hatching, 719–720

HBRUSH data type, 618

HCURSOR data type, 618

HDC data type, 618

headers

cell, customizing, 1101

functions, 233–234

programs, 45

Help > About menu, 1075–1076

helper functions

collections of objects, 761–763

lists, 768

maps, 770

highlighting cells on mouseover, 1110–1114

hInstance argument, 622

HINSTANCE data type, 618

horizontal guides, in controls, 821

hPrevInstance argument, 622

HS_BDIAGONAL hatching, 720

HS_CROSS hatching, 720

HS_DIAGCROSS hatching, 720

HS_FDIAGONAL hatching, 720

HS_HORIZONTAL hatching, 720

HS_VERTICAL hatching, 720

Hungarian notation

MFC (Microsoft Foundation Classes), 640

Windows programs, 620

iprefix, 619–620

.idb file extension, 19

IDE (Integrated Development Environment)

:: (colons), scope resolution operator, 26

assemblies, 25

Class View, 10, 17

compiler, 9

components, 9

configurations, 19

debugging, 19–23

documentation, 12–13

Editor, 10

editor, 9

error handling, 23–24

executing programs, 20–21

file extensions, 19

libraries, 9

linker, 9

namespaces, 26

options, setting, 26

Output window, 10

project folder, 13

projects

current settings, displaying, 14

defining, 13–19, 21–23

definition, 13

naming, 14

type of, displaying, 14

viewing, 16–17

projects, samples

CLR console application, 24–27

empty console application, 21–23

Win32 console application, 13–21

Property Manager, 10, 16

release versions, 19–23

Resource View, 10, 16

Solution Explorer, 10, 16

solutions

creating, 18–19

definition, 13

source code, modifying, 17–18

Standard C++ Library, 9

toolbar options, 10–12

toolbars, docking, 12

user interface, 10

Windows applications, creating and executing

MFC, 28–31

Windows Forms, 31–35

identifiers, variables, 51–52

f statement

description, 117–118

extended, 120–122

nesting, 118–120

#ifdef/#endif directives, 580–581

if-else statement, nesting, 122–124

.ilk file extension, 19

images, in cells, 1096

IMPLEMENT_SERIAL() macro, 872–873

importing symbols into DLLs, 917

#include directive, 45

inclusive OR, 84–85

incomplete class definitions, 504

incrementing variables, 74–76

Indent() function, 604–605

indexed properties, 381, 388–393

indexing arrays, 160–161

indirect base classes, 475–476, 508–511

infinite loops, 137–139

inheritance. See also classes.

abstract classes, 505–508

access control, 479–482, 489–490

base classes

accessing private members, 480–482

definition, 475–476

deriving classes from, 476–479

direct, 475–476

indirect, 475–476, 508–511

pointers to, 501–503

definition, 475

derived classes

constructors, 482–486

copy constructors, 490–495

creating, 476–479

definition, 475

pointers to, 501–503

friend classes, 495–496

incomplete class definitions, 504

multiple levels of, 508–511

pointers to class objects, 501–503

polymorphism, 501

protected class members, 486–489

inheritance (continued)

virtual destructors, 511–516

virtual functions

definition, 499

description, 497–501

pure, 505

references with, 503–504

inheritance, C++/CLI

access specifiers, 531–532

class libraries, creating, 532–535

delegates

calling, 540–541

creating, 537–541

declaring, 537

definition, 536

unbound, 541–545

derived reference classes, 524–526

derived value classes, 520–524

events

creating, 545–546

definition, 536

handling, 547–548

finalizers, 549–551

generic classes

collection classes, 555–561

defining, 551–552

`Dictionary<TKey, TValue>`, generic dictionary,
558

interface classes, 554–555

`LinkedList<T>`, generic doubly linked list, 557

`List<T>`, generic list, 556–557

using, 552–561

interface classes, 526–531

new functions, 536

reference class destructors, 549–551

visibility specifiers, 531

`InheritedStyle` property, **1100–1101**

`Initial status bar` option, **658**

initializing

multidimensional arrays, 170–172

one-dimensional arrays, 164–165

pointers, 176–179

references, 197–198

variables, 53–54

`initonly` fields, **394–395**

inline functions, in classes, **342–343**

input/output operations

`>>` (angle brackets), extraction operator, 62–63

“...” (double quotes), special character indicator, 65

C++/CLI, 104–108

command line output, 63–64, 104

escape sequences, 65–67

formatting output, 64–65, 104–107

keyboard input, 62–63, 107–108, 152–153

manipulators, 64–65

special characters, 65–67

streams, 48

`Insert()` function, **1094**

`InsertAfter()` function, **765–766, 773**

`InsertBefore()` function, **765–766, 773**

`InsertCopy()` function, **1094**

instance handles, **622**

instances

counting, 366–367

definition, 335

overview, 332–334

instantiation, **335**

`irt` data type, **54**

integer type modifiers, **56–57**

integer variables, **54–55**

Integrated Development Environment (IDE). See IDE
(Integrated Development Environment).

interface, DLL, **906, 913**

interface classes, **526–531, 554–555**

interior pointers, **225–228**

internal specifier, **531**

`InvalidateRect()` function, **731–732**

`IsEmpty()` function, **773**

J

jagged arrays, **213–216**

joining strings, **217–219**

K

keyboard input, **62–63, 107–108**

keys, databases, **923**

keywords, C++

C++/CLI, 1132

ISO/ANSI, 1131–1132

reserved, 52

L**lprefix, 619–620****late binding, 904–906****left angle brackets (<<), shift left, 86–87****left button detection, 725, 728****less than, equal sign (<=), relational operator, 115–116****less than (<), relational operator, 115–116****libraries**

definition, 9

IDE, 9

MFC (Microsoft Foundation Classes), 9

MSDN (Microsoft Development Network), 12–13

Standard C++ Library, 9

Windows Forms, 9

lifetime of variables. See storage duration.**Limits menu, 1074–1075****lines, drawing**

alternating dash-dot, 718

CLine class, 737–741

dashed, 718

dotted, 718

LineTo() function, 714–715

with the mouse, 737–741

solid, 718

straight, 714–715

LineTo() function, 714–715**linked lists, in structs, 331****LinkedList<T>, generic doubly linked list, 557****linker, IDE, 9****links, displaying in cells, 1096****list boxes**

creating, 853–855

creating the dialog class, 854–855

definition, 819

displaying, 855

illustration, 819

scaling, 852–853

Windows Forms, 1058–1061

listeners, 604**lists**

adding elements, 764–766

collection classes

AddHead() function, 764–766

adding elements, 764–766

AddTail() function, 764–766

CList template class, 763–768

CompareElements() function, 767

ConstructElements() function, 768

counting elements, 773

definition, 760

deleting elements, 767–768

DestructElements() function, 768

empty, verifying, 773

Find() function, 766–767

GetHeadPosition() function, 766

GetTailPosition() function, 766

helper functions, 768

InsertAfter() function, 765–766

InsertBefore() function, 765–766

iterating through, 766

pointers to elements, 772–773

RemoveHead() function, 767–768

RemoveTail() function, 767–768

removing elements, 772–773

searching, 766–767

searching for pointers, 773

counting elements, 773

definition, 760

deleting elements, 767–768

empty, verifying, 773

helper functions, 768

iterating through, 766

pointers to elements, 772–773

removing elements, 772–773

searching, 766–767

searching for pointers, 773

List<T>, generic list, 556–557**literal fields, in classes, 377–378. See also `initonly` fields.****literals, 59****load-time dynamic linking, 904–906****local scope, 88****logical coordinates, 789–790****logical operators, 124–127****long data type, 55****long double data type, 58****LookUp() function, 769****loops**

for, 133–139

continue statement, 139–140

definition, 132–133

do-while, 146–147

for each, 153–156

floating-point counters, 143

infinite, 137–139

multiple counters, 136–137

nesting, 147–150

non-integer counters, 140–143

while, 143–145

lp prefix, 619–620

LPARAM data type, 619

lpCmdLine argument, 622

LPCSTR data type, 619

LPHANDLE data type, 619

LRESULT data type, 619

Lvalues, 68

M

main() function

description, 46

passing arguments to, 250–252, 267–268

managed C++, 3

manipulators, 64–65

mapping modes

description, 709–711

scale factor, 844–846

scaleable, 842–844

maps, collection classes

CMap template class, 768–771

definition, 760

hashing addresses, 768–769

HashKey() function, 770–771

helper functions, 770

LookUp() function, 769

retrieving objects from, 769

storing objects in, 769

masked elements, 814–815

Maximize box option, 658

MaximizeBox property, 1068

Maximized option, 658

m_bContinuePrinting data member, 887

m_bDirect data member, 886

m_bPreview data member, 886

MDI (Multiple Document Interface). See also MFC (Microsoft Foundation Classes).

creating applications, 671–674

definition, 650

member access operators, 153

member functions

adding to a class, 339–341

defined outside the class, 362–363

defining, example, 448–453

definition, 335–336

positioning, 341–342

protecting, 361–362

static, 367–368

members of classes, 335–336

memory. See also dynamic memory allocation.

addresses, 68

leaks, 257, 594–601

menu items

adding to element menu, 686–687

adding to menu bar, 684–686

event handlers, 1049–1056

message handlers

COMMAND messages, 689

command update handler, 697–699

ContinueRouting() method, 698

creating, 688

Enable() method, 698

initializing class data members, 694–696

menu message functions, coding, 692–696

menu message functions, creating, 690–692

menu messages, 690

message types, 688–689

modifying class constructor, 695–696

SetCheck() method, 698

SetRadio() method, 698

SetText() method, 698

update handlers, 699–700

UPDATE_COMMAND_UI messages, 689

user interface update, 697–700

modifying, 687

menu message functions

coding, 692–696

creating, 690–692

menu messages, 690

menus

context

associating with classes, 795–796

checking items, 800–802

choosing, 797–802

creating, 794–795

deleting elements, 807–808

drawing highlighted elements, 806–807

highlighting elements, 802–806

identifying selected elements, 798–800

masked elements, 814–815

moving elements, 808–814

positioning elements, 813–814

servicing menu messages, 807–814

Windows Forms, 1049, 1079–1086

WM_MOUSEMOVE handler, 810–811

creating, 684–688, 1037–1038

resource files, 684

message handler definitions, 679–681**message handlers**

- COMMAND messages, 689
- command update handler, 697–699
- ContinueRouting() method, 698
- creating, 688
- Enable() method, 698
- initializing class data members, 694–696
- menu message functions, 690–692
- menu message functions, coding, 692–696
- menu message functions, creating, 690–692
- menu messages, 690
- message types, 688–689
- modifying class constructor, 695–696
- mouse, 727–729, 747–753
- SetCheck() method, 698
- SetRadio() method, 698
- SetText() method, 698
- update handlers, 699–700
- UPDATE_COMMAND_UI messages, 689
- user interface update, 697–700

message loop, 628–629**message maps, 678–681****message processing functions, 632–637****message pump, 628–629****message queue, 617****messages**

- capturing, 755–756
- categories, 681–682
- command, 682–683
- context menu, servicing, 807–814
- control notification, 681
- types of, 688–689, 725–727
- windows
 - client area, drawing, 634–636
 - cooperative multitasking, 629–630
 - decoding, 633–636
 - definition, 681
 - device content, 634–636
 - display content, 634–636
 - message loop, 628–629
 - message processing functions, 632–637
 - message pump, 628–629
 - multitasking, 629–630
 - non-queued, 627–628
 - pre-emptive multitasking, 629–630
 - queued, 627–628

methods. See member functions.**MFC (Microsoft Foundation Classes). See also MDI (Multiple Document Interface); SDI (Single Document Interface); Windows programs.**

- application class, 641–642
 - applications, creating. *See also* MDI (Multiple Document Interface); SDI (Single Document Interface).
 - application class, 653
 - basic classes, 653–654
 - CDocTemplate class, 653
 - CMyApp class, 653–654
 - CMyDoc class, 653–654
 - CMyView class, 653–654
 - CMyWnd class, 653–654
 - CSingleDocTemplate class, 653
 - development tools, 655–656
 - document class, 653
 - example, 28–31
 - frame window class, 653
 - view class, 653
 - CFrameWnd class, 642
 - CWinApp class, 641–642
 - document interfaces, 650
 - documents
 - definition, 650
 - linking to views, 652–653
 - template classes, 653
 - templates, 652–653
 - exceptions, 283
 - frame windows, 642
 - Hungarian notation, 640
 - library, 9
 - minimal application, example, 641
 - multiple top-level document architecture, 650
 - notation, 640
 - Resource Editor, 655–656
 - structure of, 640–644
 - views
 - definition, 651
 - linking to documents, 652–653
 - window class, 642
- MFC Application wizard**
- class definitions, 664–667
 - Class View tab, 661
 - classes, viewing, 663–664
 - creating MDI applications, 671–674
 - creating SDI applications
 - base classes, 660
 - Browser style toolbar option, 659

MFC Application wizard (continued)

- creating SDI applications
 - CEditView class, 660
 - CFormView class, 660
 - CHtmlEditView class, 660
 - CHtmlView class, 660
 - CListView class, 660
 - Context-sensitive help option, 659
 - CRichEditView class, 660
 - CScrollView class, 660
 - CTreeView class, 660
 - CView class, 660
 - Dialog based option, 657–658
 - Document/View architecture support option, 657–658
 - Initial status bar option, 658
 - Maximize box option, 658
 - Maximized option, 658
 - Minimize box option, 658
 - Minimized option, 658
 - Multiple top-level documents option, 657–658
 - options, 657–660
 - Printing and print preview option, 659
 - Resource language option, 657–658
 - SDI (Single Document Interface), 657–660
 - Split window option, 658
 - Standard docking toolbar option, 659
 - Thick Frame option, 658
 - Use Unicode libraries option, 657–658
- definition, 655
- executable module, creating, 667–668
- InitInstance() function, 669–671
- precompiled header files, 668
- project files, viewing, 662
- Property Pages tab, 661
- Resource View tab, 661
- Run() function, 671
- running the program, 668
- Solution Explorer tab, 661
- MFC extension, 906–907**
- Microsoft Development Network (MSDN) library, 12–13**
- Microsoft Foundation Classes (MFC). See MFC (Microsoft Foundation Classes).**
- Microsoft Intermediate Language (MSIL), 2**
- middle button detection, 728**
- Minimize box **option, 658**
- MinimizeBox **property, 1068**

- Minimized **option, 658**
- minus sign, greater than (->)**
 - indirect member access operator, 332
 - indirect member selection operator, 332
- minus signs (--), decrement operator**
 - description, 74–76
 - overloading, 426–427, 467
- MK_CONTROL flag, 728–729**
- MK_LBUTTON flag, 728–729**
- MK_MBUTTON flag, 728–729**
- MK_RBUTTON flag, 728–729**
- MK_SHIFT flag, 728–729**
- m_lpUserData data member, 887**
- MM_ANISOTROPIC mapping mode, 710–711, 842–844**
- MM_HIENGLISH mapping mode, 709–711**
- MM_HIMETRIC mapping mode, 709–711**
- MM_ISOTROPIC mapping mode, 709–711, 842–844**
- MM_LOENGLISH mapping mode, 709–711, 792–793**
- MM_LOMETRIC mapping mode, 709–711**
- MM_TEXT mapping mode, 709–711**
- MM_TWIPS mapping mode, 710–711**
- m_nCurPage data member, 887**
- m_nNumPreviewPages data member, 887**
- modal dialog boxes, 824**
- modeless dialog boxes, 824**
- modes, DataGridView control, 1092–1099**
- mouse, drawing with**
 - black-and-white, 747
 - bounding rectangles, 739–741
 - button movement, detection, 725
 - CCircle class, 744–746
 - CCurve class, 746
 - CElement class, 736–737
 - circles, 744–746
 - CLine class, 737–741
 - CreateElement() function, 751–753
 - creating elements, 751–753
 - CRectangle class, 742–743
 - Ctrl key detection, 728
 - cursor position, recording, 729–731
 - curves, 746
 - Draw() function, 738–739
 - drawing mode, 747–749
 - element classes, 732–736
 - Ellipse() function, 745–746
 - InvalidateRect() function, 731–732
 - left button detection, 725, 728
 - lines, 737–741

message handlers, 727–729, 747–753
 messages, capturing, 755–756
 messages, types of, 725–727
 middle button detection, 728
 MK_CONTROL flag, 728–729
 MK_LBUTTON flag, 728–729
 MK_MBUTTON flag, 728–729
 MK_RBUTTON flag, 728–729
 MK_SHIFT flag, 728–729
 movement, handling, 749–751
 normalized rectangles, 741
 OnMouseMove() handler, 749–751
 pen color, 747–748
 R2_BLACK mode, 747–748
 R2_COPYPEN mode, 747–748
 R2_MASKNOTOPEN mode, 748
 R2_MASKPEN mode, 748
 R2_MASKPENNOT mode, 747–748
 R2_MERGENOTOPEN mode, 747–748
 R2_MERGEOPEN mode, 748
 R2_MERGEOPENNOT mode, 747–748
 R2_NOP mode, 747–748
 R2_NOT mode, 747–748
 R2_NOTCOPYPEN mode, 747–748
 R2_NOTMASKPEN mode, 748
 R2_NOTMERGEOPEN mode, 748
 R2_NOTXORPEN mode, 748
 R2_WHITE mode, 747–748
 R2_XORPEN mode, 748
 Rectangle() function, 743
 rectangles, 739–741, 742–743
 redrawing the client area, 731–732
 ReleaseCapture() function, 755–756
 right button detection, 728
 rubberbanding, 747, 750
 SetCapture() function, 755–756
 SetROP2() function, 747–750
 Shift key detection, 728
 temporary element storage, 734–736
 update region, 731
 WM_LBUTTONDOWN message, 725–726
 WM_LBUTTONUP message, 725, 727, 753
 WM_MOUSEMOVE message, 725, 726–727
 MoveFirst() **function, 1121**
 MoveLast() **function, 1121**
 MoveNext() **function, 1121**
 MovePrevious() **function, 1121**
 m_pPD **data member, 886**
 m_rectDraw **data member, 887**

MSDN (Microsoft Development Network) library, 12–13
MSIL (Microsoft Intermediate Language), 2
 m_strFilter() **function, 958**
 m_strPageDesc **data member, 887**
multitasking, 629–630
multidimensional arrays. See also arrays.
 dynamic memory allocation, 196–197, 209–213
 passing to functions, 245–247
 pointers, 190–192
Multiple Document Interface (MDI). See MDI (Multiple Document Interface).
 Multiple top-level documents **option, 657–658**

N

n **prefix, 619–620**
named indexed properties, 388
 namespace **keyword, 96–98**
namespaces. See also using directives.
 declaring, 96–98
 definition, 26
 description, 95–96
 multiple, 98–99
 using directives, 46, 96
native C++, 2, 3
navigating data sources, 1120–1123
 nCmdShow **argument, 622**
 NDEBUG **symbol, 578–579**
nesting
 classes, 516–519
 if statements, 118–120
 if-else statements, 122–124
 loops, 147–150
 try blocks, 282–283
.NET framework, 2
 new **functions, 536**
 new **operator, 193, 284–285**
no-arg class constructors, 345–347
non-header cells, customizing, 1101–1108
non-queued messages, 627–628
normalized rectangles, 741
notation
 Hungarian, 620, 640
 MFC, 640
 Windows programs, 620
null character, arrays, 166
null pointers, 576
 NumericUpDown **control, 1068–1070, 1073**

O

.obj file extension, 19

object code, 9

object files, 9

object-oriented programming (OOP)

concepts, 473–475

definition, 335

objects

analyzing, example, 441–445

arrays of, 363–364

combining, example, 439–441

comparing, example, 438–439

creating from class templates, 431–433

declaring, 336–337

definition, 335

overview, 332–334

pointers to, 358–360, 368–371

protecting, 360–361

ODBC approach, 929–930

old-style casts, 81

OLE DB approach, 929–930

`OnActiveView()` **function, 965–966**

`OnDraw()` **function, 711–712**

one-dimensional arrays. See also arrays.

declaring, 161–164

dynamic memory allocation, 194–196

`OnMouseMove()` **handler, 749–751**

`OnUpdate()` **function, 785–787**

OOP (object-oriented programming)

concepts, 473–475

definition, 335

operator overloading. See overloading, operators.

operator precedence, 77–78

optimistic record locking, 982

ordinals, 912

output statements, 48

Output window, 10

overloading

functions, 285–288

operators

>, greater than operator, 415–417

= (equal sign), assignment operator, 418–422

-- (minus signs), decrement operator, 426–427, 467

+ (plus sign), addition operator, 423–426

++ (plus signs), increment operator, 426–427, 467

definition, 410

description, 410

example, 412–414

full operator support, 414–415

implementing, 411–412

in reference classes, 467–470

in value classes, 461–466

override keyword, 522–524

P

p prefix, 619–620

padding cells, 1100

Padding property, 1100

parameters, functions

description, 232

initializing, 277–279

protecting, 249–250

parent windows, 615

passing-by-pointer, 241–243

passing-by-reference, 247–249

passing-by-value, 240–241

.pch file extension, 19

.pdb file extension, 19

pen color, 747–748

pen objects, 717–719

pen widths, 832–833

period (.)

member access operator, 325

member selection operator, 325

pessimistic record locking, 982

plus sign (+)

addition operator, overloading, 423–426

join operator, 217

plus signs (++) , increment operator

description, 74–76

overloading, 426–427, 467

pointer notation, 244–245

pointers. See also references; tracking references.

& (ampersand), address-of operator, 173–174

* (asterisk), indirection operator, 174

arithmetic operations with, 185–187

array names as, 187–189

and arrays, 185–192

arrays of, 179–181

to base classes, 501–503

to char, 177–179

to class objects, 501–503

constant, 183–185

to constants, 183–185

declaring, 173–174

- definition, 172–173
- de-referencing, 174
- to derived classes, 501–503
- example, 178–179
- initializing, 176–179
- interior, 225–228
- with multidimensional arrays, 190–192
- notation, with multidimensional arrays, 191–192
- to objects, 358–360, 368–371
- returning values from, 254–257
- string handling, 189–190
- in structs, 330–332
- uses for, 174–176
- polymorphism, 501**
- postfix operations, 75**
- pre-compiled header files, 19**
- pre-emptive multitasking, 629–630**
- prefix operations, 75**
- preprocessor directives, 45**
- Printing and print preview **option, 659**
- printing documents**
 - cleanup, 892–893
 - CPrintInfo class, 886
 - device context, 893
 - document size, getting, 889–890
 - GetFromPage() function, 887
 - GetMaxPage() function, 887
 - GetMinPage() function, 887
 - GetToPage() function, 887
 - m_bContinuePrinting data member, 887
 - m_bDirect data member, 886
 - m_bPreview data member, 886
 - m_lpUserData data member, 887
 - m_nCurPage data member, 887
 - m_nNumPreviewPages data member, 887
 - m_pPD data member, 886
 - m_rectDraw data member, 887
 - m_strPageDesc data member, 887
 - multipage printing, 888–889
 - preparation for, 891–892
 - print data, storing, 890
 - process description, 885–886, 894–898
 - SetMaxPage() function, 887
 - SetMinPage() function, 887
- private data members, 350–353**
- private members, 350–353**
- private protected **specifier, 531**
- program statements**
 - ; (semicolon), end of statement indicator, 47
 - assignment, 47
 - compound, 49–50
 - examples, 43, 47–49
 - multi-line, 47
 - output, 48
 - statement blocks, 49–50
 - stream of input/output, 48
- program structure. See also variables.**
 - comments, 44–45
 - headers, 45
 - #include directive, 45
 - main() function, 46
 - namespaces, 46
 - preprocessor directives, 45
 - simple example, 39–44
 - using directives, 46
 - variable declarations, 47
 - whitespace, 49
- project folders, 13**
- projects**
 - current settings, displaying, 14
 - defining, 13–19, 21–23
 - definition, 13
 - naming, 14
 - samples
 - CLR console application, 24–27
 - empty console application, 21–23
 - Win32 console application, 13–21
 - type of, displaying, 14
 - viewing, 16–17
- Property Manager, 10, 16**
- protected class members, 486–489**
- protected **keyword, 486–489**
- prototype functions, 235–239**
- PS_DASH **pen style, 718**
- PS_DASHDOT **pen style, 718**
- PS_DASHDOTDOT **pen style, 718**
- PS_DOT **pen style, 718**
- PS_INSIDEFRAME **pen style, 718**
- PS_NULL **pen style, 718**
- PS_SOLID **pen style, 718**
- public **keyword, 336**
- public protected **specifier, 531**
- pure virtual functions, 505**

Q

querying a database, 938–940. See also SQL.
queued messages, 627–628

R

R2_BLACK mode, 747–748
R2_COPYPEN mode, 747–748
R2_MASKNOTOPEN mode, 748
R2_MASKPEN mode, 748
R2_MASKPENNOT mode, 747–748
R2_MERGENOTOPEN mode, 747–748
R2_MERGEOPEN mode, 748
R2_MERGEOPENNOT mode, 747–748
R2_NOP mode, 747–748
R2_NOT mode, 747–748
R2_NOTCOPYPEN mode, 747–748
R2_NOTMASKPEN mode, 748
R2_NOTMERGEOPEN mode, 748
R2_NOTXORPEN mode, 748
R2_WHITE mode, 747–748
R2_XORPEN mode, 748
radio button messages, 830–831
read-only properties, 381
record locking, 982
Record menu, disabling, 994–996
record view, database, 941–943
recordsets
 customer details, 967
 description, 937–941
 dynaset, 935–936
 second object
 customizing, 957–961
 dialog resources, creating, 954–955
 filter parameters, defining, 958–960
 filters, adding, 958
 initializing record view, 960–961
 m_strFilter() function, 958
 multiple table views, 961–966
 OnActiveView() function, 965–966
 recordset class, adding, 950–954
 switching views, 961–965
 view activation, 965–966
 view class, adding, 954–957
 viewing product orders, 966–967
 snapshot, 935–936
 sorting, 948–950

RECT struct, 330
Rectangle() function, 743
rectangles, drawing
 with the mouse
 bounding rectangles, 739–741
 CRectangle class, 742–743
 normalized rectangles, 741
 Rectangle() function, 743
 rectangles, 722, 724, 739–741, 742–743
 RECT struct, 330
recursive functions, 262–265
reference class destructors, 549–551
reference class types, defining, 378–381
reference classes
 class constructors, 549–551
 derived classes, C++/CLI, 524–526
 operator overloading, 467–470
references. See also pointers; tracking references.
 declaring, 197–198
 definition, 197
 initializing, 197–198
 to objects, 371–372
 returning values from, 258–260
 with virtual functions, 503–504
RegisterClassEx() function, 625–627
registering an ODBC database, 931–933
regular DLLs, 907
relational operators, 115–116
release versions, 19–23
ReleaseCapture() function, 755–756
remainders, calculating, 69–73
Remove() function, 1094
RemoveAll() function, 772
RemoveAt() function, 773, 1094
RemoveCurrent() function, 1121
RemoveHead() function, 767–768, 772
RemoveTail() function, 767–768, 772
reserved keywords, 52
reserved names, properties, 394
resource files, 616
Resource language option, 657–658
Resource View, 10, 16
return statement, 235
RFX_() function, 941
right angle bracket (>), OR, 84–85
right angle brackets (>>), shift right, 86–87
right button detection, 728
Rollback() function, 983–984

rolling back database update, 980–982, 982–984, 997–998

RowIndex **property, 1109**

rows, database tables

- adding to tables
 - adding orders, 1023–1028
 - DataGridView control, 1094
 - dialog switching, 1010–1014
 - order data, storing, 1019–1021
 - order entry process, 1000–1001
 - order IDs, creating, 1014–1019
 - overview, 999–1000
 - recordsets, creating, 1002
 - recordsets views, creating, 1002–1006
 - resources, adding controls, 1006–1010
 - resources, creating, 1001
 - selecting products, 1021–1023
 - setting dates, 1020–1021
- alternate, customizing, 1108
- copying, 1094
- deleting, 1094

RowsDefaultCellStyle **property, 1101**

rubberbanding, 747, 750

run-time dynamic linking, 904–906

Rvalues, 68

S

s prefix, 619–620

safe_cast **operation, 108–109**

scalar properties, 381, 382–388

scale factor

- document size, 844
- mapping mode, 844–846
- scaleable mapping modes, 842–844
- scrolling, 846–848

scaling

- list boxes, 852–853
- spin buttons, 835

scope of variables

- block, 88
- global, 91–94
- local, 88

scrollbars

- definition, 819
- illustration, 819
- setting up, 847–848

scrolling

- scale factor, 846–848
- views, 787–792

SDI (Single Document Interface). See also MFC (Microsoft Foundation Classes).

- creating applications
 - base classes, 660
 - Browser style toolbar option, 659
 - CEditView class, 660
 - CFormView class, 660
 - CHtmlEditView class, 660
 - CHtmlView class, 660
 - CListView class, 660
 - Context-sensitive help option, 659
 - CRichEditView class, 660
 - CScrollView class, 660
 - CTreeView class, 660
 - CView class, 660
 - Dialog based option, 657–658
 - Document/View architecture support option, 657–658
 - Initial status bar option, 658
 - Maximize box option, 658
 - Maximized option, 658
 - Minimize box option, 658
 - Minimized option, 658
 - Multiple top-level documents option, 657–658
 - options, 657–660
 - Printing and print preview option, 659
 - Resource language option, 657–658
 - SDI (Single Document Interface), 657–660
 - Split window option, 658
 - Standard docking toolbar option, 659
 - Thick Frame option, 658
 - Use Unicode libraries option, 657–658
- definition, 650

searching

- lists, 766–767
- one-dimensional arrays, 206–209
- pointers in lists, 773
- strings, 222–225

selection color, cells, 1099–1100

SelectionBackColor **property, 1099–1100**

SelectionForeColor **property, 1099–1100**

semantic errors, 567

semicolon (;), end of statement indicator, 47

serialization

- applying
 - to documents, 876–877
 - to element classes, 877–879
 - recording document changes, 874–876
 - to shape classes, 879–881
- CArchive class, 870–871
- for a class, 874
- CObject-based classes, 872
- DECLARE_DYNAMIC() macro, 872
- DECLARE_DYNCREATE() macro, 869, 872
- DECLARE_SERIAL() macro, 872
- default constructor, 869
- disk input/output capability, 870–871
- in document class definition, 868–869
- to documents, 876–877
- IMPLEMENT_SERIAL() macro, 872–873
- moving text, 882–883
- object types, 871
- primitive types, 871
- process description, 873–874
- Serialize() function
 - definition, 870
 - serializing documents, 876–877
 - shape classes, 879–881
- Serialize() **function**
 - definition, 870
 - serializing documents, 876–877
 - shape classes, 879–881
- SetAt() **function, 773**
- SetCapture() **function, 755–756**
- SetCheck() **method, 698**
- SetMaxPage() **function, 887**
- SetMinPage() **function, 887**
- SetRadio() **method, 698**
- SetROP2() **function, 747–750**
- SetText() **method, 698**
- shape classes, serialization, 879–881**
- shapes, collection classes, 760**
- shapes, drawing**
 - circles
 - Arc() function, 715–717
 - CCircle class, 744–746
 - Ellipse() function, 715–717
 - with the mouse, 722–723, 744–746
 - curves
 - CCurve class, 746, 775–779
 - CList template class, 774–775
 - example, 723–724

- lines
 - alternating dash-dot, 718
 - CLine class, 737–741
 - dashed, 718
 - dotted, 718
 - LineTo() function, 714–715
 - with the mouse, 737–741
 - solid, 718
 - straight, 714–715
- rectangles
 - with the mouse, 722, 724, 739–743
 - RECT struct, 330

Shift key detection, 728

short data type, 54

Show() function, 1062–1066

ShowDialog() function, 1062–1063

signed modifier data type, 56–57

Single Document Interface (SDI). See SDI (Single Document Interface).

sizeof operator, 181–183

slashes (/), comment indicator, 44

snapshot recordsets, 935–936

solid lines, 718

Solution Explorer, 10, 16

solutions

- creating, 18–19

- definition, 13

sorting

- database records, 929

- one-dimensional arrays, 205–206

- recordsets, 948–950

source code

- automatic indenting, 22

- compiler, 9

- linker, 9

- modifying, 17–18

- naming program files, 460–461

- organizing, 458–461

sources of data. See databases.

special characters, input/output, 65–67

spin buttons

- creating, 835–837

- dialog data exchange, 840

- displaying, 841–842

- initializing the dialog, 840–841

- scale dialog class, 838–841

- scale menu item, 835

- scale toolbar button, 835

- scaling, 835

- validation, 840

Split window **option**, 658

SQL

- choosing records, 925–926
- joining tables, 926–928
- retrieving data, 924–926
- sorting records, 929

stack space, 88

Standard C++ Library, 9

Standard docking toolbar **option**, 659

StartsWith() **function**, 222–225

statement blocks, 49–50

static

- constructors, 396
- controls, 819
- data members, 365–367
- member functions, 367–368
- properties, 393–394
- storage duration, 91, 94–95
- variables, 261–262

static **specifier**, 94–95

static_cast **keyword**, 80–81

status bars

- adding to frames, 848–850
- overview, 848
- parts of, 850–851
- updating, 851–852

storage duration, 88–91, 94–95

streams, input/output, 48

string handling

- dynamic memory allocation, 216–225
- multidimensional arrays, 171–172
- one-dimensional arrays, 166–169
- pointers, 189–190

strings

- array termination character, 166
- C++/CLI
 - + (plus sign), join operator, 217
 - description, 216
 - EndsWith() function, 222–225
 - joining, 217–219
 - modifying, 220–221
 - searching, 222–225
 - StartsWith() function, 222–225
 - trim() function, 220–221
 - trimming, 220–221
- counting characters, 154–156, 189–190
- extracting substrings, 305–307, 319–320

- looping through, 154–156
- removing blanks, 294–295, 316

structs

- . (period)
 - member access operator, 325
 - member selection operator, 325
- > (arrow)
 - indirect member access operator, 332
 - indirect member selection operator, 332
- defining, 324–325
- definition, 324
- example, 326–329
- initializing, 325
- intellisense assistance, 329–330
- linked lists, 331
- members access, 325–329, 330–332
- pointers, 330–332
- RECT, 330
- unions, 410

Style **property**, 1101

styles, cells

- default, 1101–1108
- dynamic, 1108–1114

submenus, Windows Forms, 1038–1040

switch **statement**, 129–131

synonyms for data types, 60

syntactic errors, 567

system menu, 616

System **using directive**, 1033–1034

System::Collections **using directive**, 1033–1034

System::Collections::IList **interface**, 1114

System::ComponentModel **using directive**, 1033–1034

System::ComponentModel::BindingList **interface**, 1114

System::ComponentModel::BindingListView **interface**, 1114

System::ComponentModel::IListSource

interface, 1114

System::Data **using directive**, 1033–1034

System::Data::DataColumn **class**, 1090

System::Data::DataRow **class**, 1090

System::Data::DataSet **class**, 1090

System::Data::DataTable **class**, 1090

System::Drawing **using directive**, 1033–1034

System::Windows::Forms **using directive**, 1033–1034

sz **prefix**, 619–620

T

tab controls, Windows Forms, 1040–1042

`TabControl` control, **1040–1042, 1127–1128**

table adapters, 1090

tables, databases

adding rows to

adding orders, 1023–1028

dialog switching, 1010–1014

order data, storing, 1019–1021

order entry process, 1000–1001

order IDs, creating, 1014–1019

overview, 999–1000

recordsets, creating, 1002

recordsets views, creating, 1002–1006

resources, adding controls, 1006–1010

resources, creating, 1001

selecting products, 1021–1023

setting dates, 1020–1021

multiple, 1127–1128

tag names, 408–409

template classes, 653

template member functions, 430–431

templates, MFC documents, 652–653

temporary element storage, 734–736

ternary operator. See conditional operator.

`Text` menu item, **859–860**

`Text` property, **1068**

`Thick Frame` option, **658**

`this` pointer, **358–360**

throwing exceptions

C++/CLI, 308–309

description, 281

example, 280–281

tilde (~), NOT, 86

title bar, 615

toolbar, navigation, 1120–1123

toolbars

adding buttons to

editing button properties, 701–702

overview, 700–701

testing the buttons, 703

tooltips, 703–704

IDE, 10–12

tooltips, 703–704

`ToString()` function, **376–377**

`TotalNumberOfItems()` function, **1121**

`Trace` class, **602–611**

trace switches, 605–607

tracepoints, 572–573

tracking handles, 199–200

tracking references, 225. See also references.

transactions, database update, 982–984

`trim()` function, **220–221**

trimming strings, 220–221

trivial scalar properties, 384–388

true/false data types, 57

try blocks, 281–283

typed pointers, collection classes

`AddHead()` function, 772

`AddTail()` function, 772

`CTypedPtrList` class, 771

`CTypedPtrList` operators, 771–773

`Find()` function, 773

`FindIndex()` function, 773

`GetAt()` function, 772

`GetCount()` function, 773

`GetHead()` function, 772

`GetHeadPos()` function, 773

`GetNext()` function, 772

`GetPrev()` function, 772

`GetTail()` function, 772

`GetTailPos()` function, 773

`InsertAfter()` function, 773

`InsertBefore()` function, 773

`IsEmpty()` function, 773

`RemoveAll()` function, 772

`RemoveAt()` function, 773

`RemoveHead()` function, 772

`RemoveTail()` function, 772

`SetAt()` function, 773

typedef keyword, 60

type-safe collections, 760–761

U

unbound delegates, 541–545

unbound mode, 1093–1099

unboxing data types, 101

unconditional branching, 132

`Unindent()` function, **604–605**

unions

anonymous, 409–410

in classes and structures, 410

defining, 408–409

definition, 407

tag names, 408–409

unmanaged C++, 3**Update()** function, 980–982**update handlers, 699–700****update mode**

- button label, changing, 992–993
- Cancel button, visibility, 993–994
- canceling the update, 997–998
- description, 990–991
- edit controls, enabling/disabling, 991–992
- expediting the update, 996–997
- Record menu, disabling, 994–996

update region, 731**UpdateAllViews()** function, 785–787**UPDATE_COMMAND_UI messages, 689****updating databases**

- adding records, 980–982
- adding rows to tables
 - adding orders, 1023–1028
 - dialog switching, 1010–1014
 - order data, storing, 1019–1021
 - order entry process, 1000–1001
 - order IDs, creating, 1014–1019
 - overview, 999–1000
 - recordsets, creating, 1002
 - recordsets views, creating, 1002–1006
 - resources, adding controls, 1006–1010
 - resources, creating, 1001
 - selecting products, 1021–1023
 - setting dates, 1020–1021
- AddNew() function, 980–982
- BeginTrans() function, 983–984
- canceling updates, 980–982, 997–998
- CancelUpdate() function, 980–982
- CDatabase class, 983–984
- CommitTrans() function, 983–984
- completing an update, 980–982
- controls
 - button label, changing, 992–993
 - Cancel button, visibility, 993–994
 - canceling the update, 997–998
 - description, 990–991
 - edit controls, enabling/disabling, 991–992
 - expediting the update, 996–997
 - Record menu, disabling, 994–996
- CRecordset operations, 980–982
- Delete() function, 980–982
- deleting records, 980–982
- Edit() function, 980–982

- editing records, 980–982
- example, 984–988
- inhibiting update, 988–990
- optimistic record locking, 982
- pessimistic record locking, 982
- record locking, 982
- Rollback() function, 983–984
- rolling back an update, 982–984
- starting an update, 988–990
- tables, adding rows to
 - adding orders, 1023–1028
 - dialog switching, 1010–1014
 - order data, storing, 1019–1021
 - order entry process, 1000–1001
 - order IDs, creating, 1014–1019
 - overview, 999–1000
 - recordsets, creating, 1002
 - recordsets views, creating, 1002–1006
 - resources, adding controls, 1006–1010
 - resources, creating, 1001
 - selecting products, 1021–1023
 - setting dates, 1020–1021
- transactions, 982–984
- Update() function, 980–982
- update mode
 - button label, changing, 992–993
 - Cancel button, visibility, 993–994
 - canceling the update, 997–998
 - description, 990–991
 - edit controls, enabling/disabling, 991–992
 - expediting the update, 996–997
 - Record menu, disabling, 994–996
 - validating operations, 981

Use Unicode libraries option, 657–658**user interface. See GUI (Graphical User Interface).****using directives, 46, 96, 1033–1034****V****validating**

- database update operations, 981
- input, dialog boxes, 1063–1066

value classes

- defining, 373–378
- derived classes, C++/CLI, 520–526
- deriving, 520–526
- overloading, 461–466

Value **property, 1109**

variables

See also `const` modifier

See also literals

See also pointers

See also storage duration

See also unions

assigning values to, 47

automatic variables, 88–91

casting

in assignment statements, 79–80

definition, 78

explicit casts, 80–81

old-style casts, 81

rules for, 78–79

debugging, 576–578

declaring, 47, 52–53, 91

decrementing, 74–76

default type, 88–91

definition (declaring), 53

definition (meaning of), 51

enumeration, 60–62

identifiers, 51–52

incrementing, 74–76

initialization, 53–54

limiting values of, 60–61

modifying, 73–74

naming, 51–52, 619–620

positioning declaration, 91

prefixes, 619–620

reserved keywords, 52

scope

block, 88

global, 91–94

local, 88

stack space, 88

static storage duration, 91, 94–95

storage duration, 88–91, 94–95

vertical bars (|), logical OR, 125–126

view class, 653, 711–712, 954–957

viewing data. See DataGridView control.

views. See also documents.

client coordinates, 789–792

creating elements, 833–834

logical coordinates, 789–790

MM_LOENGLISH MAPPING MODE, 792–793

OnUpdate() function, 785–787

scrolling, 787–792

UpdateAllViews() function, 785–787

updating multiple, 785–787

virtual class destructors, 511–516

virtual destructors, 511–516

virtual functions

definition, 499

description, 497–501

inheritance, 497–501

pure, 505

references with, 503–504

virtual keyword, 499–501

virtual mode, 1093

visibility specifiers, 531

W

w prefix, 619–620

watch, setting, 576

wchar_t data type, 56

Web browser controls, 1047–1048

WebBrowser control, 1047–1048

while loops, 143–145

whitespace, 49

wide character data type, 56

Win32 console applications, 6, 13–21

WINAPI() function, 622

window class, registering, 625–627

WindowProc() function, 632–637

windows

client area, 708

creating, 625–627

elements

borders, 615

child windows, 615

client area, 615

control menu, 616

illustration, 615

parent windows, 615

resource files, 616

system menu, 616

title bar, 615

ending the program, 636

example program, 637–638

initializing, 627

look and feel, 622

messages

client area, drawing, 634–636

cooperative multitasking, 629–630

decoding, 633–636

definition, 681

- device content, 634–636
- display content, 634–636
- message loop, 628–629
- message processing functions, 632–637
- message pump, 628–629
- multitasking, 629–630
- non-queued, 627–628
- pre-emptive multitasking, 629–630
- queued, 627–628
- specifying, 623–625
- Windows Forms**
- containers, 1031
- definition, 1031
- dialog boxes
 - adding, 1068–1075
 - button clicks, handling, 1076–1079, 1083–1086
 - ComboBox control, 1068–1070, 1072
 - context menu, responding to, 1079–1086
 - ControlBox property, 1068
 - creating, 1056–1062
 - DialogResult property, 1057, 1063–1066
 - event handler, Reset menu, 1067–1068
 - FormBorderStyle property, 1068
 - getting data from, 1070–1073
 - Help > About menu, 1075–1076
 - input controls, disabling, 1073–1074
 - Limits menu, 1074–1075
 - MaximizeBox property, 1068
 - MinimizeBox property, 1068
 - NumericUpDown control, 1068–1070, 1073
 - Show() function, 1062–1066
 - ShowDialog() function, 1062–1063
 - Text property, 1068
 - using, 1062–1068
 - validating input, 1063–1066
- library, 9
- overview, 1031–1032
- Windows Forms, application databases**
- accessing data, 1090–1091
- AddNew() function, 1121
- BindingNavigator control, 1120–1123
- BindingSource component, 1115–1120
- bound mode, 1114–1115
- cells
 - alignment, 1100
 - background color, 1099–1100
 - buttons, 1095
 - CellStyle property, 1109
 - ColumnIndex property, 1109
 - default styles, 1101–1108
 - DesiredType property, 1109
 - dynamic styles, 1108–1114
 - fonts, 1099–1100
 - foreground color, 1099–1100
 - formatting, 1100
 - FormattingApplied property, 1109
 - header, customizing, 1101
 - highlighting on mouseover, 1110–1114
 - images, 1096
 - non-header, customizing, 1101–1108
 - padding, 1100
 - properties, 1109
 - RowIndex property, 1109
 - selection color, 1099–1100
 - Value property, 1109
 - wrapping text, 1100
- columns
 - adding, 1096
 - binding to controls, 1123–1127
 - buttons in cells, 1095
 - check boxes, 1095
 - defining, 1093–1099
 - drop-down lists, 1096
 - editing, 1096–1098
 - images in cells, 1096
 - links, displaying, 1096
 - width, setting, 1098–1099
- controls
 - appearance, 1102–1103
 - binding to columns, 1123–1127
 - binding to data sources, 1115–1120
 - column format, 1106–1108
 - column headers, 1105–1106
 - setting up, 1103–1104
 - CurrentPosition() function, 1121
- data source classes, 1090
- DataGridView control
 - Add() function, 1094
 - AddCopy() function, 1094
 - bound mode, 1093
 - Clear() function, 1094
 - column types, 1095–1099
 - DataGridViewButtonColumn column, 1095
 - DataGridViewCheckBoxColumn column, 1095
 - DataGridViewComboBoxColumn column, 1096
 - DataGridViewImageColumn column, 1096

Windows Forms, application databases (continued)

Windows Forms, application databases (continued)

- DataGridView control
 - DataGridViewLinkColumn column, 1096
 - definition, 1090
 - displaying data, 1091–1093
 - Insert() function, 1094
 - InsertCopy() function, 1094
 - modes, 1092–1099
 - Remove() function, 1094
 - RemoveAt() function, 1094
 - unbound mode, 1093–1099
 - virtual mode, 1093
- DataGridView control, customizing
 - Alignment property, 1100
 - AlternatingRowsDefaultCellStyle property, 1101
 - BackColor property, 1099–1100
 - DataGridViewCellStyle object, 1099–1100
 - DefaultCellStyle property, 1101
 - Font property, 1099–1100
 - ForeColor property, 1099–1100
 - Format property, 1100
 - InheritedStyle property, 1100–1101
 - Padding property, 1100
 - RowsDefaultCellStyle property, 1101
 - SelectionBackColor property, 1099–1100
 - SelectionForeColor property, 1099–1100
 - Style property, 1101
 - System::Collections::IList interface, 1114
 - System::ComponentModel::BindingList interface, 1114
 - System::ComponentModel::BindingListView interface, 1114
 - System::ComponentModel::IListSource interface, 1114
 - WrapMode property, 1100
- displaying data, 1090–1091
- MoveFirst() function, 1121
- MoveLast() function, 1121
- MoveNext() function, 1121
- MovePrevious() function, 1121
- navigating data sources, 1120–1123
- RemoveCurrent() function, 1121
- rows
 - adding, 1094
 - alternate, customizing, 1108
 - copying, 1094
 - deleting, 1094

- System::Data::DataColumn class, 1090
- System::Data::DataRow class, 1090
- System::Data::DataSet class, 1090
- System::Data::DataTable class, 1090
- TabControl control, 1127–1128
- table adapters, 1090
- tables, multiple, 1127–1128
- toolbar, navigation, 1120–1123
- TotalNumberOfItems() function, 1121
- updating a database, 1122–1123

Windows Forms, application GUI

- Button control, 1044–1046
- context menus, 1049
- ContextMenuStrip control, 1049
- dialog boxes
 - adding, 1068–1075
 - button clicks, handling, 1076–1079, 1083–1086
- ComboBox control, 1068–1070, 1072
- context menu, responding to, 1079–1086
- ControlBox property, 1068
- creating, 1056–1062
- DialogResult property, 1057, 1063–1066
- event handler, Reset menu, 1067–1068
- FormBorderStyle property, 1068
- getting data from, 1070–1073
- Help > About menu, 1075–1076
- input controls, disabling, 1073–1074
- Limits menu, 1074–1075
- MaximizeBox property, 1068
- MinimizeBox property, 1068
- NumericUpDown control, 1068–1070, 1073
- Show() function, 1062–1066
- ShowDialog() function, 1062–1063
- Text property, 1068
- using, 1062–1068
- validating input, 1063–1066

- dialog button events, 1058–1060
- dialog objects, creating, 1061–1062
- event handler, dialog button, 1058–1060
- event handlers, menu items, 1049–1056
- FixedDialog property, 1056–1062
- FormBorderStyle property, 1056–1062
- GroupBox control, 1042–1044
- grouping controls, 1042–1044
- list boxes, 1058–1061
- menu items, event handlers, 1049–1056
- menus, 1037–1038
- overview, 1035–1036

submenus, 1038–1040
 tab controls, 1040–1042
 TabControl control, 1040–1042
 testing, 1048–1049
 Web browser controls, 1047–1048
 WebBrowser control, 1047–1048

Windows Forms, applications

creating, 31–35
 example, 645–647
 overview, 1032–1034
 properties, modifying, 1034–1035
 starting, 1035
 System using directive, 1033–1034
 System::Collections using directive, 1033–1034
 System::ComponentModel using directive,
 1033–1034
 System::Data using directive, 1033–1034
 System::Drawing using directive, 1033–1034
 System::Windows::Forms using directive,
 1033–1034
 using directives, 1033–1034

Windows programs. See also MFC (Microsoft Foundation Classes).

API (application programming interface), 617–618
 b prefix, 619–620
 BOOLEAN data type, 618
 by prefix, 619–620
 BYTE data type, 618
 c prefix, 619–620
 C++ standards, 5–6
 C++ versions, 5
 CHAR data type, 618
 CLR console applications, 6
 command line pointer, 622
 console applications, 6
 data types, 618–619
 dw prefix, 619–620
 DWORD data type, 618
 event-driven, 7, 617
 events, 7
 fn prefix, 619–620
 h prefix, 619–620
 HANDLE data type, 618
 handles, 622
 HBRUSH data type, 618
 HCURSOR data type, 618
 HDC data type, 618
 HINSTANCE data type, 618

Hungarian notation, 620
 i prefix, 619–620
 instance handles, 622
 l prefix, 619–620
 lp prefix, 619–620
 LPARAM data type, 619
 LPCSTR data type, 619
 LPHANDLE data type, 619
 LRESULT data type, 619
 message queue, 617
 n prefix, 619–620
 operating system, 616
 organizing, 638–640
 p prefix, 619–620
 programming concepts, 6–8
 s prefix, 619–620
 structure of. See also WindowProc() function;
 WinMain() function.
 required functions, 620–621
 sz prefix, 619–620
 variable name notation, 619–620
 variable prefixes, 619–620
 w prefix, 619–620
 Win32 console applications, 6
 window class, registering, 625–627
 window elements
 borders, 615
 child windows, 615
 client area, 615
 control menu, 616
 illustration, 615
 parent windows, 615
 resource files, 616
 system menu, 616
 title bar, 615
 WindowProc() function, 632–637
 windows
 creating, 625–627
 ending the program, 636
 example program, 637–638
 initializing, 627
 look and feel, 622
 specifying, 623–625
 windows messages
 client area, drawing, 634–636
 cooperative multitasking, 629–630
 decoding, 633–636
 device content, 634–636

Windows programs (continued)

Windows programs (continued)

- windows messages
 - display content, 634–636
 - message loop, 628–629
 - message processing functions, 632–637
 - message pump, 628–629
 - multitasking, 629–630
 - non-queued, 627–628
 - pre-emptive multitasking, 629–630
 - queued, 627–628
- WORD data type, 619
- WinMain() **function**
 - arguments, 622
 - example, 631–632
 - hInstance argument, 622
 - hPrevInstance argument, 622
 - lpCmdLine argument, 622
 - nCmdShow argument, 622
 - prototype, 621
- RegisterClassEx() function, 625–627
- WINAPI() function, 622
- WNDCLASSEX struct, 623–625
- wizard. See MFC Application wizard.**
- WM_LBUTTONDOWN **message, 725–726**
- WM_LBUTTONUP **message, 725, 727, 753**
- WM_MOUSEMOVE **handler, 810–811**
- WM_MOUSEMOVE **message, 725, 726–727**
- WM_PAINT **message, 708, 711, 731, 754–755**
- WNDCLASSEX **struct, 623–625**
- WORD data type, 619**
- WrapMode **property, 1100**
- wrapping text, 1100**
- Write() **function, 103–104, 603–604**
- WriteIf() **function, 603–604**
- WriteLine() **function, 103–104, 603–604**
- WriteLineIf() **function, 603–604**
- write-only properties, 381**

