

# Index

## A

**abstract classes, 539–542**

**abstract keyword, virtual functions in C++/CLI, 556**

**access control**

access levels for inherited class members, 522–523

class inheritance and, 513–514

classes, 344

**access specifiers, for classes and interfaces, 566**

**ActiveX Data Objects (ADO), 1120–1121**

**Add Event Handler, 816**

**Add Member Variable Wizard, 820–821**

**addition operators**

concatenating strings, 473–476

overloading, 432–436

**addition operators, C++/CLI, 494**

**AddNew(), CRecordset update operations, 1168–1169**

**addresses, returning, 265–266**

**address-of operators, 175–176**

**ADO (ActiveX Data Objects), 1120–1121**

**AFX\_EXT\_CLASS keyword, 1108**

**algorithms, 674–676**

copy(), 669–672

fill(), 674

find(), 675

overview of, 605

replace(), 674

sort(), 623–624

transform(), 675–676

**Alt+F9, setting breakpoints, 701**

**American National Standards Institute (ANSI), 5.**

*See also ISO/ANSI C++*

**American Standard Code for Information**

**Interchange (ASCII)**

characters, 53–54

code, 1309–1315

**analytical operators, 451–452**

**AND bitwise operators, 80–81**

**AND operators, 127**

**animate control, 988**

**anonymous unions, 419**

**application class (CWinApp), 772**

**application start-up, Windows Forms, 835**

**Application Verifier option, Debug menu, 703**

**Application Wizard**

in automatically generated console programs, 47–48

creating applications, 28–30, 748

creating Win32 console applications, 14–19

creating Windows Forms applications, 30–33

creating simple C++ programs, 38

defined, 9

DLLs and, 1095

file menu print options, 1064

location of project files, 790–791

message maps, 806–807

print preparation and, 1070

serialization built into, 1048

tools for creating MFC-based Windows programs, 784

**applications, writing, 3–4**

**Arc(), CCircle class, 855, 884**

**arguments**

main(), 258–260, 275–276

omitting, 286–287

passing to functions, 247–251

pointers as, 283–285

**arithmetic**

calculations, 65–70

expressions, defined, 45

expressions, evaluating, 323–325

function objects for arithmetic operations, 673

pointers, 187–189

**arithmetic operators, 449–451**

**arrays, 161–206**

address-of operators, 175–176

buffer arrays, 300

char, 179–181

character arrays and string handling, 168–169

## arrays (continued)

### arrays (continued)

- constant pointers and pointers to constants, 185–187
- counting characters, 191–192
- declaring, 163–166
- defined, 162–163
- delete operators, 195
- dynamic memory allocation and, 194–199
- exercises, 238
- for arrays, 195–196
- free store, using, 196–198
- free store vs. heap, 194
- helper functions, 913
- indirect data access. See indirect data access
- indirection operators, 176
- initializing, 166–167
- initializing pointers, 178–179
- library functions for strings, 200–206
- multidimensional. See multidimensional arrays
- multiple data values of same type, 162–174
- names as pointers, 189–191
- new operators, 195
- null-terminated strings, 200–206
- of objects, 371–372
- overview, 161, 911–912
- passing to functions, 251–253
- pointer arithmetic and, 187–189
- of pointers, 181–183, 285
- pointers, declaring, 175
- pointers, generally, 174, 187–192
- pointers, working with, 176–178
- programming with strings, 170–171
- references, declaring and initializing, 199–200
- sizeof operators, 183–185
- storing multiple strings, 173–174
- string input, 169–171
- summary, 236–237

**arrays, C++/CLI, 206–238**

- arrays of arrays, 221–223
- CLR, 209–213
- comparing strings, 229
- exercises, 238
- interior pointers, 233–237
- joining strings, 224–225
- modifying strings, 227–229
- multidimensional, 217–221
- one-dimensional, searching, 215–217
- one-dimensional, sorting, 213–214
- overview, 206–207
- searching strings, 230–233

- sorting associated, 214
- strings in, 224–227
- summary, 236–237
- tracking handles, 207–209
- tracking references, 233

### ASCII (American Standard Code for Information Interchange)

- characters, 53–54
- codes, 1309–1315

### assemblies

- C++/CLI programming language, 565
- defined, 25

### assertions

- C++/CLI debugging, 735–737
- native C++ debugging, 708–709

### assign()

- deleting elements from a list, 634
- replacing contents of a vector, 618

### assignment operators

- implementing for C++/CLI reference types, 503
- overloading, 427–431

### assignment statements

- calculations, 65
- casting, 77–78
- defined, 45

### associated arrays, C++/CLI, 214

### associative containers, 651–665

- defined, 606
- map containers. See map containers, STL/CLR
- multimap containers, 664–665
- multiset containers, 651
- overview of, 651
- set containers, 651

### associative containers, STL/CLR, 685–692

### at(), accessing elements in vectors, 615–616

### Attach to Process option, Debug menu, 704

### attributes

- defined, 6
- GDI (Graphical Device Interface), 849

### Audio Video Interleaved (AVI), 988

### automatic variables, in storage duration and scope, 89

### Automation, MFC extension DLLs, 1100

### Autos window, debugger, 705–707

### AVI (Audio Video Interleaved), 988

## B

### back(), accessing elements in vectors, 616

### bad pointers, 263–264

Base **address, hash values and, 919–920**

**base classes**

- accessing private members of base class in a derived class, 513–516
- classed derived from, 510–511
- indirect, 542–544
- overriding functions in, 570
- overview of, 509–510
- pointers to, 535–537
- references to, 537
- specifying virtual functions, 532–533

begin(), **random access iterators and, 609**

BEGIN\_MESSAGE\_MAP(), **806–807**

BeginTrans(), CDatabase **transaction operations, 1171–1172**

**binary predicates, 606**

**binary serialization, CLR, 1078–1079**

**binding, to individual controls, 1299–1302**

BindingNavigator **component**

- overview of, 1296
- in System::Windows::Forms namespace, 1267
- updating a database, 1297–1298
- working with, 1296–1297

BindingSource **component, 1290–1296**

- overview of, 1290–1291
- in System::Windows::Forms namespace, 1267
- working with, 1291–1296

**bitmap images, adding to toolbar buttons, 842–843**

**bitwise operators, 79–86**

- AND, 80–81
- exclusive OR, 83
- NOT, 84
- OR, 82–83
- overview, 79
- shift operators, 84–86

**block scope, of automatic variables, 86**

**blocks, 47**

**body of functions**

- defined, 44
- overview, 242–243

**Boolean variables, 55**

BooleanSwitch, **switching debugging/tracing on/off, 735–736**

**borders, window elements, 745**

**bound mode, DataGridView controls. See DataGridView controls, in bound mode**

**bounding rectangles, CLine class, 877–880**

**boxing/unboxing class type values, C++/CLI programming language, 554**

**breakpoints**

- advanced, 701–704

disabling, 701

setting, 700–701

tracepoints, setting, 702–703

**brushes**

- creating, 859
- creating in CLR, 1039
- interior color created with, 855
- using, 860

**buddy controls, 1001**

**buffer arrays, 300**

**bugs**

- common, 697–698
- defined, 695
- finding next, 722
- program, 696–697

**Build Solution**

- building DLLs, 1105
- Ctrl+Shift+B (Build), 797

Button **controls**

- Windows Forms, 831
- working with, 1224–1226

**buttons**

- click events, 1238–1239
- click-handler, 1253–1256
- controlling visibility of, 1180–1181
- labels, 1180

## C

**C++/CLI programming, defined, 5–6**

**calculations, 65–76**

- arithmetic operations as, 65–70
- assignment statements, 65
- const modifiers in, 67–68
- constant expressions in, 68
- displaying results of, 70
- for input from keyboards, 68–69
- lvalues in, 65
- modifying variables, 71–72
- operator precedence in, 74–76
- overview, 65
- for remainders, 70–71
- for results, 69–70
- rvalues in, 65
- sequence of, 74–76
- using increment and decrement operators, 72–74

**calculators, 299–315**

- analysis for, 300–302
- assembling program for, 309–311
- blanks from strings, eliminating, 302
- expressions, evaluating, 303–305

## calculators (*continued*)

---

### calculators (*continued*)

- extending program for, 311–312
- functions in, 303
- modified programs for, 314–315
- numbers analysis, 306–309
- overview, 299–300
- substrings, extracting, 312–314
- value of terms, 305–306

### calculators, C++/CLI programming, 322–328

- arithmetic expressions, 323–325
- input strings, removing spaces from, 323
- numbers evaluation, 325–326
- substrings, extracting, 327–328
- value of terms, 325

### call stack, debugging programs and, 715–716

#### cancel button

- controlling visibility, 1180–1181
- implementing cancel operation, 1185

CancelUpdate(), CRecordset **update operations, 1168–1169**

capacity(), vector containers, **611**

CArchive **class, 1050–1052**

CArray **template class**

- helper functions, 913
- overview of, 911–912

**case labels, 132**

**case values, 131**

**cases, sharing, 133–134**

**<cassert> header, 708**

**casting, 76–79**

- assignment statements, 77–78
- explicit, 78–79
- old-style, 79
- operands, rules for, 76–77

catch **block, 289–291**

**Category ID field, in relational database application, 1115**

CBox **class, 508–509**

CBox **class**

- adding data members, 456–457
- adding function members, 458–462
- adding global functions, 463–465
- addition operators, overloading, 432–436
- analyzing CBox objects, 451–452
- basic operations, 447
- combining CBox objects, 449–451
- comparing CBox objects, 448–449
- constructor, 457
- defining, 455–456

- implementing, 447–448
- multifile project, 453–455
- simple destructor, 410–412
- using, 465–468

CBox **objects**

- analyzing, 451–452
- combining, 449–451
- comparing, 448–449
- operator function and, 421

CBrush **objects**

- creating, 859
- setting color with, 855
- using, 860

CButton **class, 987**

CCircle **class**

- constructor for, 883–884
- drawing circles, 884
- implementing, 883
- overview, 882

CClientDC **class, 852, 888**

CCmdUI, **member functions, 825**

CCurve **class**

- defining, 925–927
- drawing, in a window, 884–885
- exercising, 925–927
- implementing, 927–928

CCustomerView **dialog, 1205–1207**

CDatabase **class, 1128, 1171–1172**

CDate **class, 1207**

CDC **class**

- circles, 854–856
- displaying graphics, 852–853
- drawing modes, 886
- lines, 853–854
- overview of, 852

CDialog **class**

- adding, 990–992
- developing, 1258–1260
- for edit boxes, 1023–1024
- for list boxes, 1019–1020
- overview of, 986

CDocument **class, 780**

CElement **class. See also elements**

- adding pen widths to elements, 999–1000
- class definitions, 874–875
- CText derived from, 1026
- defining element classes, 870–873
- getting elements to move themselves, 961

- serializing element classes, 1048, 1058–1059
- storing temporary elements, 873–874
- cell styles**
  - DataGridView cell style properties, 1275–1276
  - dynamically setting for DataGridView control, 1284–1289
  - headers, customizing, 1276–1277
  - nonheaders, customizing, 1277
- CellFormatting **events, 1284–1289**
- CFile **object, CArchive class and, 1050**
- CFont **class, 1030**
- CFrameWnd **class, 773**
- char, **179–181**
- char **array, 412**
- character**
  - ASCII codes, 1309–1315
  - data types, 53–54
- character arrays, string handling an, 168–169**
- child windows, window elements, 745**
- Choose **menu item**
  - click event, 1260–1263
  - dealing with, 1257
- Circle **class, defining circles for CLR drawings, 903–904**
- circles**
  - defining for CLR drawings, 903–904
  - drawing, 854–856, 884
- class constructors, 351–358**
  - adding to CBOX class, 351–353
  - C++/CLI programming, 557
  - calling, 516–519
  - CCircle class, 883–884
  - CLine class, 875
  - CRectangle class, 881
  - defaults, 353–355
  - defined, 351–358
  - in derived classes, 516–519
  - initializing, 358
  - modifying, 823
  - parameter values, assigning, 355–356
  - parameter values, supplying, 356–357
  - working with, 351–353
- class definitions**
  - CElement class, 874–875
  - message handlers, 807–809
  - serialization in document class and, 1048–1049
- class destructors**
  - for CMessage class, 413–415
  - default destructor, 410–412
  - overview of, 409–410
- class hierarchy, multiple levels of, 544**
- class inheritance, 507–599**
  - access control, 513–514
  - access levels, for inherited class members, 522–523
  - accessing private members of base class, 514–516
  - base classes, 509–510
  - constructors, in derived classes, 516–519
  - copy constructors, in derived classes, 524–528
  - defined, 509
  - derived classes, 510–513
  - exercises, 597–599
  - friend classes, 530
  - multiple levels of, 542–544
  - native C++ compared with C++/CLI, 558
  - OOP basic concepts, 507–509
  - overview of, 507
  - protected members, 519–522
  - summary, 596–597
  - virtual functions
    - working with inherited functions, 531–532
- class inheritance, C++/CLI, 555–558**
  - access specifiers, 566
  - boxing/unboxing class type values, 554
  - class libraries, creating, 566–568
  - class libraries, using, 568–570
  - classes, generic, 585–588
  - classes and assemblies and, 565
  - collection classes, generic, 590–595
  - delegates, calling, 574–576
  - delegates, creating, 571–574
  - delegates, declaring, 571
  - delegates, overview, 571
  - delegates, unbound, 576–579
  - derived reference classes, 558–560
  - destructors and finalizers, in reference classes, 583–585
  - event handling, 581–583
  - events, 571
  - events, creating, 580–581
  - exercises, 597
  - functions, specifying new, 570–571
  - interface classes, defining, 561–562
  - interface classes, generic, 589–590
  - interface classes, implementing, 562–565
  - overview of, 553–554
  - summary, 596–597
  - virtual functions, 555–558
  - visibility specifiers, 565
- class interface, 446–447**

## class libraries

### **class libraries, C++/CLI**

- creating, 566–568
- using, 568–570

### **class members**

- access levels for inherited, 522–523
- as friends, 528–529, 530
- interface classes, 561
- private, 513
- protected, 519–522
- public, 513

### **class objects**

- pointers to, 535–537
- storing in vector containers, 618–622

### **class properties, C++/CLI, 390–403**

- indexed properties, complex, 401–402
- indexed properties, defining, 397–398
- indexed properties, working with default, 398–401
- overview, 390–403
- reserved property names, 403
- scalar properties, defining, 390–393
- scalar properties, working with, 393–397
- static properties, 402–403
- trivial scalar properties, 393

### **class templates**

- defining, 438–440
- member functions, 440–441
- with multiple parameters, 444–446
- objects created from, 441–444
- overview of, 438

### **Class View, defining CBox class, 455–456**

### **Class View tabs, 17**

### **Class Wizard, adding dialog class with, 990**

### **classes, 343–351**

- access control, 344
- adding to MFC extension DLLs, 1103–1104
- addition operators, overloading, 432–436
- anonymous unions, 419
- arrays of objects of, 371–372
- assignment operators, overloading, 427–431
- casting between, 550
- CBox class, adding data members, 456–457
- CBox class, adding function members, 458–462
- CBox class, adding global functions, 463–465
- CBox class, analyzing CBox objects, 451–452
- CBox class, combining CBox objects, 449–451
- CBox class, comparing CBox objects, 448–449
- CBox class, constructor for, 457
- CBox class, defining, 455–456
- CBox class, implementing, 447–448
- CBox class, multifile project with, 453–455
- CBox class, using, 465–468

class definitions, 793–797

class interface, 446–447

class templates, 438

class templates, defining, 438–440

class templates, member functions, 440–441

class templates, objects created from, 441–444

class templates, with multiple parameters, 444–446

CMessage class, 413–415

comparison operators, overloading, 424–426

const objects of, 368–371

constructors, 351

constructors, modifying, 823

context menus, associating with, 945–946

copy constructors, 415–417

data members of, 345–347

declaring objects of, 344–345

default destructor, 410–412

defining, 344

destructors, 409–410

document templates, 783

dynamic memory allocation, 412–413

exercises, 504–505

first, 340–342

incomplete definitions, 538

increment and decrement operators, overloading, 436–437

inline functions, 350–351

member function definitions, positioning, 349–350

member functions of, 347–349

menu messages, 817

nested classes, 550–553

operations on, 342–343

operator overloading, 420–423

overview, 340–343

overview of, 409

pointers to objects of, 376–379

private members, 358–366

program code, file names, 470–471

program code, organizing, 468

references to objects of, 376–379

serialization, 1054

serialization, preparing for, 1080–1081

static members, 373–376

STL templates, 601

strings, accessing/modifying, 476–480

strings, comparing, 480–484

strings, concatenating, 473–476

strings, creating string objects, 471–473

strings, overview, 471

strings, searching, 484–489

- strings, sorting words from text, 489–494
- summary, 503–504
- terminology of, 343
- `this` pointer, 366–368
- `typeid()`, for determining class type, 1063
- unions, between variables, 417–419
- unions, in classes and structures, 419
- viewing, 792–793
- Windows applications, 783–784
- classes, C++/CLI, 494, 565–568**
  - access specifiers for, 566–568
  - assignment operators, implementing, 503
  - decrement operators, overloading, 499–500
  - exercises, 504–505
  - reference classes, overloading, 500–502
  - reference types, implementing, 503
  - summary, 503–504
  - value classes, overloading, 494–498
  - visibility specifiers for, 565
- `ClassView` **context menu, 785, 792–793**
- `ClassWizard`, 806
- `clear()`, **deleting elements from a list, 633**
- CLI (Common Language Infrastructure), 2**
- click events**
  - buttons, 1238–1239, 1253–1256
  - Choose menu item, 1260–1263
  - context menus, 1256–1257
  - input validation, 1242–1244
- client area**
  - drawing, 765–766
  - drawing in a window and, 848
  - redrawing, 758, 869–870
  - window elements, 745
- client coordinates**
  - dealing with, 940–941
  - overview of, 939
  - pixels for measuring, 1009
- `CLine` **class**
  - bounding rectangles, 877–880
  - constructor for, 875
  - drawing lines, 875
  - implementing, 875
  - `Move()`, 961
  - normalized rectangles, 880
  - overview of, 875
- `CList` **class, 913–918**
  - adding elements to a list, 913–915
  - deleting objects from a list, 917
  - drawing a curve, 924–925
  - helper functions for a list, 917–918
  - iterating through a list, 916
  - overview of, 913
  - searching a list, 916–917
  - working with, 923
- CLR (Common Language Runtime)**
  - arrays, 209–213
  - calculator program for, 322–328
  - console applications, 7
  - console projects, 24–26
  - debugging CLR programs, 731
  - defined, 2–3
  - fruity console program, 100–102
  - function arguments, variable numbers of, 275
  - heap, 381–385
  - nested `if` statements, 153–154
  - standards and, 5–6
  - Windows Forms program executing CLR project, 775–777
  - writing C++ applications and, 3–4
- CLR sketch application**
  - context menus, 976–977
  - Element Move operation, 979
  - highlighting elements for selection, 972–976
  - moving elements, 979–981
- `CMap` **class, 918–920**
  - helper functions, 919–920
  - overview of, 918–919
- `CMenu` **class, 950**
- `CMessage` **class**
  - class destructor for, 413–415
  - copy constructors, 415–417
- `CMultiDocTemplate`, 783
- `CMyApp` **class, Windows application classes, 783–784**
- `CMyDoc` **class, Windows application classes, 783–784**
- `CMyView` **class, Windows application classes, 783–784**
- `CMyWnd` **class, Windows application classes, 783–784**
- `CObject` **class**
  - as base class for element classes, 870
  - functionality of, 1052
  - serialization and, 1049
- code**
  - ASCII characters, 1309–1315
  - closing dialog boxes, 994–995
  - debugging code, 709–714
  - displaying dialog boxes, 994
  - object code, 9
  - source code, 17–19
  - Unicode characters, 1315

## Code Definition windows

### Code Definition windows, 10–11 collections, 910

- CArray template class, 911–912
- CCurve class, defining, 925–927
- CCurve class, exercising, 925–927
- CCurve class, implementing, 927–928
- CList template class, drawing a curve, 924–925
- CList template class, overview, 913–918
- CList template class, using, 923
- CMap template class, 918–920
- CTypedPtrList template class, 921–923
- generic C++/CLI classes, 590–595
- helper functions, 913
- overview of, 909
- typed pointer, 920
- types of, 910
- type-safe, 911

### collocation, sorting words from text, 489

### color drawings, 857–860

- brushes, creating, 859
- brushes, using, 860
- defining color for CLR drawings, 899
- pens, creating, 857–858
- pens, using, 858–859

### Color menu, adding to menu bar, 815

### columns, DataGridView

- formats, 1281–1283
- headers, 1280–1281
- types of, 1272

### columns, System::Data::DataColumn namespace, 1266

### COM, 1120–1121

### combo box controls, 1035–1037

### comma operators, 73–74

### command messages

- adding data members to class definitions, 820–821
- creating menu message functions, 817–820
- message categories and, 809–810
- MFC handling, 810
- overview of, 816–817
- processing, 810–811

### command-line arguments

- accessing, 276
- receiving, 259–260

### comments, in programs, 42

### CommitTrans(), CDatabase transaction operations, 1171–1172

### common controls

- Button controls, 1224
- overview of, 988
- WebBrowser control, 1226

### Common Language Infrastructure (CLI), 2

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

### Common Type System (CTS), 3

### comparison operators

- comparing CBox objects, 448–449
- full support for, 424–426
- pair objects and, 653
- strings, 480–484
- testing extended class in program debugging,  
720–722
- using with SQL SELECT statements, 1117

### comparisons, function objects for, 672–673

### comparisons, values. See values, comparing

### compilation time, precompiled headers and, 797–798

### compiler, ISO/ANSI C++, 455

### compound statements, 47

### concatenating strings, 473–476

### conditional operators, 130–131

### configurations, 20

### console applications

- automatically generated, 47–48
- files created by building, 19
- overview of, 6–7

### const modifiers

- in calculations, 67–68
- using, 257–258

### const objects, of classes

- member functions and, 369–371
- overview, 368–369

### constant expressions, 68

### constant pointers, 185–187

### constructors. See class constructors

- defining for CBox class, 457
- new operator and, 412

### container adapters, 603

### containers. See also associative containers;

#### sequence containers

- container adapters, 603
- double-ended queue containers, 626–629
- list containers. See list containers, STL/CLR
- map containers. See map containers, STL/CLR
- multimap containers, 664–665
- multiset containers, 651
- overview of, 602–603
- priority queue containers, 643–648
- queue containers. See queue containers
- set containers, 651
- stack containers, 649–651
- vector containers. See vector containers

**containers, STL/CLR**

- defining curves, 967–968
- double-ended queue containers, 680–684
- list containers, 684–685
- overview of, 677

**context menus**

- adding to Windows Forms applications, 1228
- associating with a class, 945–946
- choosing, 946–948
- creating in CLR, 976–977
- exercising pop-ups, 949–950
- highlighting elements for selection, 952–956
- identifying selected element, 948–949
- implementing, 943–945
- menu items, checking, 950–952
- overview of, 943
- responding to, 1256–1257

ContextMenuStrip **controls, 1228**

**continue statements, 142–143****control menu, window elements, 745****control notification messages, 809–810****control variables, 1004****controls**

- adding to dialog boxes, 988–990
- adding to dialog resources, 1193–1196
- ASCII codes for, 1309–1315
- binding to, 1299–1302
- common controls, 988
- DataGridView control, in unbound mode, 1279–1280
- dialog controls, 1248–1251
- edit boxes. *See* edit box controls
- edit controls, enabling/disabling, 1178–1180
- exercises, 1045
- initializing, 996–997
- input controls, disabling, 1251
- linking to recordsets, 1136–1138
- list boxes. *See* list box controls
- overview of, 986–987
- spin button control. *See* spin button control
- status bars. *See* status bar controls
- summary, 1044–1045
- tab sequence, 1004
- types of, 986–987
- variables, 1004

**controls, CLR**

- combo boxes, 1035–1037
- drawing text, 1038
- exercises, 1045
- fonts, choosing, 1039–1040
- fonts, creating, 1039

- overview of, 1030–1031
- summary, 1044–1045
- text boxes, 1042–1043
- text elements, creating, 1037, 1043
- TextElement class, 1041

Controls **property, GroupBox controls, 1233**

**cooperative multitasking, Windows programs, 759**

**coordinate system transformations, CLR, 965–967**

**coordinates**

- comparing logical, client, and screen, 1009
- logical coordinates and client coordinates, 939–941
- printing documents and, 1075–1077

copy(), **669**

**copy constructors**

- default, 364–366
- implementing, 379–380, 415–417
- reference class types, 389–390

**copy constructors, in derived classes, 524–528**

- fixing, 526–527
- how it works, 525
- overview of, 524–526

**copying null-terminated strings, 203–204**

count(), **accessing objects in a map, 655**

**counting characters, 191–192**

## CPen objects

- creating pens, 857–858
- overview of, 857
- using pens, 858–859

CPoint **class, 924**

.cpp **file, 806**

CPrintInfo **class, 1066–1068**

CProductView **class, 1132–1133**

Create(), **992**

CreateElement(), **889, 1000–1001**

CreateWindow(), **756**

CRecordset **class**

- checking legality of update operations, 1169
- creating recordsets and, 1128
- database update operations, 1168–1169
- record locking, 1170
- sorting, 1138–1139

CRecordView **class**

- creating, 1144–1147
- initializing, 1149–1150
- overview of, 1132–1134

CRectangle **class**

- constructor for, 881
- drawing rectangles, 881–882

\_crtDbgFlag, **free store debugging, 725–726**

crtDBG.h, **723**

## \_CrtSetReportFile()

`_CrtSetReportFile()`, **free store debugging**,  
**726–727**

`_CrtSetReportMode()`, **free store debugging**,  
**726–727**

`CScaleDialog` **class**, **1005, 1018**

`CScrollView` **class**, **1012**

`CSingleDocTemplate` **class**, **783**

`CSketcherDoc` **class**, **820–821**

`CSScrollView` **class**, **935–937**

`CStatic` **class**, **987**

`CStatusBar` **class**, **1014**

`CString` **class**, **1024–1025**

`CstringW` **class**, **1130**

`CText` **class**

constructor for, 1027–1028

derived from `CElement` class, 1026

drawing `CText` object, 1028

implementing, 1027

moving `CText` object, 1028

`CTextDialog` **class**, **1023**

`Ctrl+Alt+X` (**Toobox window**), **836**

`Ctrl+D` (**tab sequence**), **1004**

`Ctrl+F5` (**Execute**), **797–798**

`Ctrl+Shift+B` (**Build**), **797**

`Ctrl+Shift+N` (**New Project**), **785, 1100**

`Ctrl+T` (**test**), **990**

**CTS (Common Type System)**, **3**

`CTypedPtrList` **class**, **921–923**

creating sketch document with, 929–931

functions, 921–923

overview of, 921

**current position, graphics displayed relative to**, **852**

`Curve` **class**, **967–968**

**curve collections**, **884–885**

defining, 925–927

exercising, 925–927

implementing, 927–928

**curves**

drawing with `CList` template class, 924–925

STL/CLR containers for defining, 967–968

**customer recordset**

adding, 1156–1157

filtering, 1160–1163

**customer resource dialog**

creating, 1157–1158

linking to order dialog, 1163–1165

**customer view class**, **1158–1160**

`CView` **class**

scrolling views, 937–939

`UpdateAllViews()`, 935–937

views in frame windows, 781

`CWinApp` **class**, **772**

## D

**data**

corruption, common bugs, 697

retrieving with SQL, 1116–1117

setting up for `DataGridView` control, 1278–1279

**data access, indirect. See indirect data access**

**data exchange, spin button control**, **1006**

**data members**

adding for `CBox` class, 456–457

adding to class definitions, 820–821, 1230–1231

`CPrintInfo` class, 1066–1067

defined, 341

dialog box support, 995–996

initializing new, 821–823

nonserializable, 1079

**data sources. See database applications**

**data sources, accessing from Windows Forms. See**

**Windows Forms, data source access**

`Data Sources` **window**, **1266**

**data types, 51–60**

Boolean variables, 55

character, 53–54

class constructors, 351

for enumeration constants, 109–110

floating-point numbers, 55–56

integer variables, 52–53

in ISO/ANSI C++, 56–57

literals, 57–58

multiple data values of same type. **See multiple data values, of same type**

options for use in loops, 143–145

overview, 51–52

synonyms for, 58

type modifiers, 54–55

variables with specific sets of values, 58–60

Windows, 748–749

**data types, C++/CLI, 97–112**

enumerations, 107–112

fruity CLR console program and, 100–102

input from keyboards, 106–107

output, formatting, 103–106

output to the command line, 102

overview, 97–100

`safe_cast`, using, 107

value class types, 98

**data types, defining, 331–380**

- arrays of class objects, 371–372
- classes, operations on, 342–343
- classes, understanding, 343–351
- const objects, 368–371
- exercises, 406–407
- first class, 340–342
- indirect member selection operators, 340
- IntelliSense and, 337–338
- objects, 340–342
- overview, 331
- pointers, to class objects, 376–379
- pointers, to structs members, 339–340
- private members, of classes, 358–366
- references to class objects, 376–379
- static members, of classes, 373–376
- structs, accessing members of, 333
- structs, initializing, 333
- structs, overview of, 332–333
- structs, working with, 334–337
- structures generally, 332–340
- summary, 405–406
- terminology in, 343
- this pointer, 366–368

**data types, defining in C++/CLI, 381–407**

- class properties. *See* class properties, C++/CLI
- exercises, 406–407
- initonly fields, 403–404
- literal fields, 385–386
- overview, 381
- reference class types, copy constructors for, 389–390
- reference class types, defining, 386–387
- reference class types, working with, 387–389
- static constructors, 405
- summary, 405–406
- ToString() function in classes, 384–385
- value class types, defining, 381–382
- value class types, working with, 383–384

**database applications**

- creating, 1122
- customer recordset, adding, 1156–1157
- customer recordset, filtering, 1160–1163
- customer resource dialog, creating, 1157–1158
- customer resource dialog, linking to order dialog, 1163–1165
- customer view class, creating, 1158–1160
- data transfer between database and recordset, 1131–1132
- database viewer, exercising, 1165–1166

- exercises, 1166
- exercising example, 1138
- MFC supported, 1120–1121
- multiple table views, accessing, 1150–1151
- ODBC classes, 1121–1122
- ODBC database, registering, 1122–1124
- ODBC program, generating, 1124–1125
- overview of, 1113–1116
- product orders, viewing, 1151–1155
- queries, 1129–1131
- record view class, adding, 1143–1147
- record view class, initializing, 1149–1150
- record view class, overview, 1132–1134
- recordset class, adding, 1141–1143
- recordsets, creating, 1127–1129
- recordsets, filtering, 1147–1148
- recordsets, linking controls to, 1136–1138
- recordsets, sorting, 1138–1139
- recordsets, using second recordset object, 1140–1141
- snapshot vs. dynaset records, 1126
- summary, 1166
- view dialog, creating, 1134–1136
- views, switching, 1151–1155
- window caption, modifying, 1139–1140

**databases, SQL. *See* SQL (Structured Query Language)****databases, update operations**

- BindingNavigator component, 1297–1298
- button labels, changing, 1180
- cancel button, visibility of, 1180–1181
- cancel operation, 1185
- CDatabase transaction operations, 1171–1172
- checking legality of operations, 1169
- controlled updating, 1185–1186
- CRecordset supporting, 1168–1169
- customizing application for, 1173–1174
- dialog resources, adding controls, 1193–1196
- dialog switching, implementing, 1197–1200
- edit controls, enabling/disabling, 1178–1180
- example of, 1172–1173
- exercises, 1215
- expediting the update, 1183–1184
- how it works, 1174–1175
- implementing update mode, 1177–1178
- managing, 1175–1177
- order data, storing, 1205–1207
- order entry process, 1186–1187
- Order IDs, creating, 1200–1205
- orders, adding new, 1209–1214

## databases (continued)

### databases (continued)

- overview of, 1167–1168
- products, selecting, 1207–1209
- record locking and, 1169–1170
- Record menu, disabling, 1181–1183
- recordset views, creating, 1188–1193
- recordsets, creating, 1188
- resources, creating, 1187–1188
- summary, 1214
- tables, adding rows to, 1186–1187
- transactions, 1170
- DataGridView **controls**
  - modes, 1268–1269
  - overview of, 1267–1268
  - in System::Windows::Forms namespace, 1267
- DataGridView **controls, in bound mode**
  - binding to individual controls, 1299–1302
  - BindingNavigator component, 1296–1297
  - BindingSource component, 1290–1296
  - overview of, 1269, 1290
  - tables, working with multiple, 1302–1304
  - updating database, 1297–1298
- DataGridView **controls, in unbound mode, 1269–1275**
  - cell styles, dynamically setting, 1284–1289
  - column formats, 1281–1283
  - column headers, setting up, 1280–1281
  - control, setting up, 1279–1280
  - customizing, 1275–1277
  - data, setting up, 1278–1279
  - overview of, 1268
  - rows, customizing alternate, 1283–1284
- DataSet class, in System::Data namespace, **1266**
- dates, setting, 1207**
- DDV (Dialog Data Validation), 1006**
- DDX (Dialog Data Exchange), 1006**
- Debug classes, C++/CLI**
  - overview of, 732
  - working with, 737–741
- Debug menu, 703–704**
- \_DEBUG preprocessor symbol, 709–710**
- Debug toolbar, 701–703**
- debugger**
  - basic debugging operations, 699–700
  - overview of, 698
  - starting, 700
- debugging code, adding, 709–714**
- debugging techniques, 695–731**
  - assertions, 708–709

- basic operations, 698–700
- breakpoints, advanced, 701–704
- breakpoints, setting, 700–701
- common bugs, 697–698
- debugging code, adding, 709–714
- debugging dynamic memory. *See* dynamic memory allocation
- debugging programs. *See* programs, debugging overview of, 695–696
- program bugs, 696–697
- starting debugging, 703–705
- summary, 741–742
- tracepoints, setting, 702–703
- variable values, changing, 707
- variable values, inspecting, 705–707
- debugging techniques, C++/CLI**
  - assertions, 736
  - Debug and Trace, classes, 732
  - Debug and Trace, working with, 737–741
  - output, controlling, 735–736
  - output, generating, 732–733
  - output, indenting, 734
  - output destination, 734
  - overview of, 731–732
  - summary, 741–742
  - trace output in Windows Forms applications, 741
- decimals, ASCII codes, 1309–1315**
- decision making, 117–153. *See also* loops**
  - comparing values, 117–119
  - conditional operators, 130–131
  - continue statements, 142–143
  - data types, options for use in loops, 143–145
  - do-while loops, 148–149
  - exercises, 159–160
  - extended if statements, 122–124
  - floating-point loop counters, 145–146
  - if statements, 119–120
  - indefinite for loops, 140–141
  - logical operators and expressions, 126–129
  - for loop, using, 136–137
  - for loop, variations on, 138
  - multiple counters, using, 139–140
  - nested if statements, 120–122
  - nested if-else statements, 124–126
  - nested loops, 149–152
  - overview, 117
  - repeating blocks of statements generally, 135
  - summary, 159
  - switch statements, 131–134

- unconditional branching, 134
- while loops, 146–148
- decision making, C++/CLI. See also loops, C++/CLI**
- decisions making, C++/CLI, 153–160**
  - for each loops, 156–159
  - exercises, 159–160
  - nested if statements, 153–154
  - overview, 153
  - summary, 159
- declarations**
  - arrays, 163–166
  - defined, 43–44
  - delegates, C++/CLI, 571
  - friend classes, 530
  - namespaces, 94–96
  - object classes, 344–345
  - pointers, 175
  - pointers to functions, 280–283
  - references, 199–200
  - tracking handles, 207–209
  - variables, 50–51, 89
- `DECLARE_DYNAMIC()`, **serialization macro, 1052**
- `DECLARE_DYNCREATE()`, **serialization macro, 1049, 1052**
- `DECLARE_MESSAGE_MAP()`, **806–807**
- `DECLARE_SERIAL()`, **serialization macro, 1052–1053**
- decorated names, DLLs, 1104**
- decrement operators ( — )**
  - overloading, 436–437
  - overloading in C++/CLI, 499–500
  - working with, 72–74
- default destructor, 410–412**
- defaults**
  - class constructors, 353–355
  - copy constructors, 364–366
  - indexed properties, 397
  - mapping mode, 850
  - parameter values for constructor arguments,
    - supplying, 356–537
  - parameter values in classes, assigning, 355–356
  - type modifiers, 54
- delegates, C++/CLI**
  - calling, 574–576
  - creating, 571–574
  - declaring, 571
  - event handlers as, 838
  - overview of, 571
  - unbound, 576–579
- `Delete()`, `CRecordset` **update operations, 1168–1169**
- delete operators, 195**
- deque. See double-ended queue containers**
- derived classes**
  - accessing private members of base class in, 513–516
  - constructors in, 516–519
  - copy constructors in, 524–528
  - defined, 509
  - deriving from base class, 510–511
  - pointers to, 535–537
  - specifying new functions, 570–571
  - working with, 511–513
- derived classes, C++/CLI**
  - overview of, 553
  - reference classes, 555, 558–560
- Design window, 976**
- destructors**
  - calling wrong, 545–548
  - MFC documents, 931
  - virtual, 544
- destructors, C++/CLI, 583–585**
- destructors, class**
  - default destructor, 410–412
  - overview of, 409–410
- device context**
  - GDI (Graphical Device Interface), 849
  - linking device-independent Windows API output functions to drivers and devices, 765–766
  - preparing for printing documents, 1072–1073
- device coordinates. See client coordinates**
- Device Points to Logical Points (DPTOL), 940–941**
- dialog boxes**
  - buttons added to, 1187–1188
  - code for closing, 994–995
  - code for displaying, 994
  - controls added to, 988–990, 1193–1196
  - creating, 988
  - customer resource dialog, 1157–1158
  - data members supporting, 995–996
  - dialog class (CDialog), adding, 990–992
  - dialog class (CDialog), developing, 1258–1260
  - dialog form, creating, 1257–1258
  - displaying, 992–993, 1021
  - elements, creating in view, 1000–1001
  - exercises, 1045
  - for file operations, 1086
  - initializing, 996–997, 1006–1007

## dialog boxes (continued)

### dialog boxes (continued)

- linking order dialog to customer resource dialog, 1163–1165
- modal and modeless, 992
- overview of, 985–986
- pen dialog example, 1001
- pen width dialog, adding to document, 998–999
- pen width dialog, adding to elements, 999–1000
- radio buttons, message handling for, 997–998
- summary, 1044–1045
- switching between, 1197–1200
- testing, 988–990
- view dialog, for database, 1134–1136
- for viewing records, 1143–1144
- dialog boxes, CLR**
  - adding, 1031
  - customizing, 1031–1032
  - displaying, 1032–1033, 1043
  - exercises, 1045
  - overview of, 1030–1031
  - pen widths, setting, 1033–1035
  - summary, 1044–1045
  - text dialog, creating, 1042
- dialog boxes, Windows Forms application**
  - adding second, 1245–1248
  - creating, 1235–1236
  - handling button events, 1238–1239
  - using, 1240–1242
- dialog class (CDialog)**
  - adding, 990–992
  - developing, 1258–1260
  - for edit boxes, 1023–1024
  - for list boxes, 1019–1020
- dialog controls, getting data from, 1248–1251**
- Dialog Data Exchange (DDX), 1006**
- Dialog Data Validation (DDV), 1006**
- dialog forms, creating, 1257–1258**
- dialog objects, creating, 1239–1240**
- direct base classes, 509**
- direct member selection operators, 345**
- `DispatchMessage()`, **retrieving Windows messages, 759–760**
- display content, linking Windows API output functions to drivers and devices, 765–766**
- .dll file, 1107**
  - `dllexport` attribute, 1108
  - `DllImport`, 1110–1111
  - `DLLMain()`, 1098, 1102–1103
- DLLs (Dynamic Link Libraries), 1093–1112**
  - contents of a DLL file, 1098

- `DLLMain()`, 1102–1103
- exporting symbols from, 1109–1111
- exporting variables and functions from, 1108
- extension DLLs, adding classes to, 1103–1104
- extension DLLs, applying in Sketcher application, 1106–1107
- extension DLLs, building, 1105
- extension DLLs, exporting classes from, 1104–1105
- extension DLLs, writing/using, 1100–1102
- files required in order to use, 1107
- how they work, 1095–1096
- importing symbols into a program, 1109–1110
- interface, 1098
- `main()`, 1098
- overview of, 1093–1095
- runtime dynamic linking, 1097–1098
- SDI applications and, 787
- types of, 1098–1099
- when to use, 1099–1100
- dockable toolbars, 12–13**
- docking controls, 1279–1280**
- document, recording document changes, 1055**
- document class, serialization in, 1048–1050**
- document objects, altering, 1055**
- document size**
  - printing documents and, 1068–1069
  - scale factor and, 1009–1010
- document storage. See serialization**
- document templates**
  - classes, 783
  - overview of, 782–783
- document views**
  - logical coordinates and client coordinates, 939–941
  - `MM_LONGENGLISH` mapping mode, 941–943
  - scrolling views, 937–939
  - updating multiple views, 935–937
- documentation, 13**
- documents, MFC**
  - adding elements, 933–934
  - creating with `CTypedPtrList` template class, 929–931
  - document destructors, implementing, 931
  - drawing sketch document, 931–933
  - exercising sketch document, 934–935
  - interfaces, 780
  - linking documents and views, 781–782
  - overview of, 780
  - templates, 782–783
- documents, printing. See printing documents**

**document/view concept, in MFC, 780**DoDataExchange(), **1006, 1136**DoFieldExchange(), **1131**DoPreparePrinting(), **1070****double-ended queue containers, 626–629****double-ended queue containers, STL/CLR, 680–684**do-while loops, **148–149**DPtoLP() (**Device Points to Logical Points**),  
**940–941**

Draw()

CLine class, 877

coordinate system transformations and, 966

Graphics class in CLR, 899–900

highlighting elements for selection, 952–956

MFC sketch document, 931–933

**drawing, in a window, 847**

brushes, creating, 859

brushes, using, 860

CCircle class, constructor for, 883–884

CCircle class, drawing circles, 884

CCircle class, implementing, 883

CCircle class, overview of, 882

CCurve class, 884–885

CDC class, 852

CElement class, 874–875

circles, 854–856

client area, overview, 848

client area, redrawing, 869–870

CLine class, bounding rectangles, 877–880

CLine class, constructor for, 875

CLine class, drawing lines, 875

CLine class, implementing, 875

CLine class, normalized rectangles, 880

color drawings, 857

CRectangle class, constructor for, 881

CRectangle class, drawing rectangles, 881–882

deleting and moving shapes, 943

device context, 849

displaying graphics, 852–853

drawing mode, selecting, 886–887

elements, creating, 889–891

elements, defining classes for, 870–873

elements, storing temporary, 873–874

exercises, 907

GDI (Graphical Device Interface), 848–849

lines, calculating enclosing rectangle for, 880

lines, overview, 853–854

mapping modes, 849–851

mouse, drawing with, 861–862, 867–869

mouse, messages from, 864–865

mouse, programming, 863–864

mouse message handlers, 865–867, 885

mouse messages, capturing, 893–895

mouse properties, 865

OnDraw() member function, 851–852

OnMouseMove() handler, 887–889

overview of, 847–848

pens, creating, 857–858

pens, using, 858–859

running the Sketcher example, 892–893

summary, 906–907

View class, 851

WM\_LBUTTONDOWN messages, dealing with, 891–892

**drawing, in a window (CLR)**

circles, defining, 903–904

colors, defining, 899

element class, defining, 897–898

exercises, 907

forms, drawing on, 895

lines, defining, 898–899

lines, drawing, 899–900

mouse event handlers, adding, 895–897

MouseMove event handler, implementing, 904–905

MouseDown event handler, implementing, 905

overview of, 895

Paint event handler, implementing, 905–906

pens, defining, 900–902

pens, standard, 902

rectangles, defining, 902–903

summary, 906–907

**drawing mode, selecting, 886–887****drawing text, CLR controls and, 1038**DrawString(), Graphics class, **1038****drivers, ODBC, 1121****dynamic memory allocation, 194–199**

class destructors and, 409

classes, 412–413

for arrays, 195–196

free store, using, 196–198

heap, 194

multidimensional arrays, 198–199

new and delete operators, 195

overview, 194

**dynamic\_cast operator, 550****dynaset records, vs. snapshot, 1126****E**eatspaces() function, **302–303**Edit(), CRecordset update operations,  
**1168–1169**

## edit box controls

### edit box controls

- creating, 1022–1023
- CString class, 1024–1025
- CText class, 1027–1028
- dialog class for, 1023–1024
- overview of, 1021
- text element, creating, 1028–1029
- text element, defining, 1026–1027
- text menu item, adding, 1025–1026

### edit controls, enabling/disabling, 1178–1180

### Editor windows, 10–11

#### Element class, C++/CLI, 897–898

#### Element menu

- adding event handlers to, 838
- adding menu items to, 814–815
- completing, 815
- positioning, 813

### Element Move operation, implementing in CLR, 979

### elements, 1000–1001

- adding to documents, 933–934
  - adding to list collections, 913–915
  - class definitions, 870–873
  - creating, 889–891
  - deleting, 957–958, 978
  - getting elements to move themselves, 961–963
  - hash values and, 919–920
  - highlighting, 952–956
  - identifying selected, 948–949
  - masked, 964–965
  - moving, 958–959
  - moving in CLR, 979–981
  - serializing element classes, 1058–1059
  - storing temporary, 873–874
  - text element, defining, 1026–1027
- Ellipse(), CCircle class, 855, 884
- ### empty console projects, 21–23
- ### encapsulation, defined, 343
- end(), random access iterators and, 609
- END\_MESSAGE\_MAP(), 806–807
- ### enum classes, visibility specifiers, 565
- ### enumerations, C++/CLI, 107–112
- defining, 58, 108–109
  - as flags, 111–112
  - native, 112
  - operations on constants, 110–111
  - overview, 107–109
  - types for constants, specifying, 109–110
  - values for constants, specifying, 110
- erase()
- deleting elements from a list, 633–634

- deleting elements from a map, 655
- deleting elements from a multimap container, 664
- deleting elements in a vector, 617

### errors

- dealing with, 23–24
- stepping over to errors, 716–720
- syntactic and semantic, 697

### escape sequences, 63–64

### Event Handler Wizard

- adding message handlers to user interface, 824–825
- creating menu message functions, 817–820
- for menu messages, 815–816

### event handlers

- implementing, 839–840
- for menu messages, 815–816

### event handlers, Windows Forms application

- creating, 1228–1229
- for Limits menu, 1235, 1252–1253
- for menu items, 1229–1230
- for Play menu item, 1231–1235

### event-driven programming

- defined, 7–8
- Windows OSs, 746

### events, 7–8

### events, C++/CLI

- creating, 580–581
- handling, 581–583
- overview of, 571

### exceptions

- catching, 290–291
- defined, 287–288
- throwing, 289
- working with, 288–289

### exclusive OR operators, 83

### .exe file extension, 1095

### executable modules, creating and running in MFC, 797–801

### Execute (Ctrl+F5), 797–798

### executing program versions, 20–21

### explicit boxing, 554

### explicit casting, 69, 78–79

### exporting classes, from extension DLLs, 1104–1105

### expr() function, 303–305

### extended if statements, 122–124

### extension DLLs

- adding classes, 1103–1104
- applying in Sketcher application, 1106–1107
- building, 1105
- exporting classes, 1104–1105

types of DLLs, 1099  
 Windows Sockets and, 1099  
 writing/using, 1100–1102  
**extract () function, 312–314**  
**extraction operators (>>), 60, 1051**

## F

**F10 (Step Over), stepping over to errors, 716–720**  
**fields**

defined, 332, 343  
 nonserializable, 1079–1081  
 in relational databases, 1114

**file extensions, options for DLLs, 1095**

**file names, program code, 470–471**

**file operations**

creating dialogs, 1086  
 implementing for a sketch, 1086–1088  
 open, 1087–1088  
 overview of, 1086  
 print, 1089–1090  
 save, 1086–1087  
 save as, 1088

**file scope, 89**

**FileMode enumeration values, 1081–1082**

**files, required in order to use DLLs, 1107**

**fill (), algorithms, 674**

**filters/filtering**

customer recordset, 1160–1163  
 defining filter parameters, 1148–1149  
 list containers, 636–639  
 recordsets, 1147–1148

**finalizers, C++/CLI reference classes, 583–585**

**find ()**

algorithms, 675  
 objects in a map, 654–655  
 objects in a multimap container, 654–655  
 strings, searching, 484–489

**Find (), searching list collections, 916–917**

**FindIndex (), searching list collections, 917**

**first class, 340**

**flags, enumerations as, 111–112**

**floating toolbars, 12**

**floating-point loop counters, 145–146**

**floating-point numbers, 55–56**

**fonts**

choosing, 1039–1040  
 creating, 1039  
 defining, 1030

for each **loops, 156–159**

for **loops**

indefinite, 140–141  
 using, 136–137  
 variations on, 138

**foreign keys, in relational databases, 1115**

Form Design **capability, 1228**

Form1 **class**

adding members, 1230–1231  
 creating dialog objects, 1239–1240

Format **property, columns, 1281–1283**

**forms. See also Windows Forms applications**

defined, 831  
 drawing in CLR drawings, 895

**frame windows**

MFC window class, 773  
 views appearing in, 781

**free store**

vs. heap, 194  
 using, 196–198

**free store, debug operations**

checking, 723–724  
 controlling, 725–726  
 output, 726–727

**friend, of a class, 528–529**

**friend classes, 530**

**friend functions**

definitions, placing, 364  
 overview, 362–364  
 working with, 362–364

**front (), accessing elements in vectors, 616**

**fruity CLR console program, 100–102**

**function adapters, 606**

**function headers, 241–242**

**function members, 343**

adding for CBox class, 458–462

**function objects**

for arithmetic operations, 673  
 for comparisons, 672–673  
 overview of, 605–606

**functional notation, 51**

**functions**

addresses returned by, 265–266  
 arguments, omitting, 286–287  
 arguments, pointers as, 283–285  
 arguments, to `main ()`, 258–260  
 arguments, variable numbers of, 260–262,  
 274–275  
 arguments passed to, 247–251  
 arrays of pointers to, 285  
 arrays passed to, 251–253

## functions (continued)

### functions (continued)

`assert()`, 708–709  
 bad pointers returning, 263–264  
 body of, 242–243  
 calculator example. See *calculators*  
 coding for menu messages, 820  
 command-line arguments, accessing, 276  
 command-line arguments, receiving, 259–260  
`const` modifiers, 257–258  
`CTypedPtrList` template class, 921–923  
 defined, 36–37  
 exceptions, catching, 290–291  
 exceptions, defined, 287–288  
 exceptions, throwing, 289  
 exceptions, working with, 288–289  
 exercises, 277–278  
 exporting from DLLs, 1108  
 headers, 241–242  
 inherited, 531–532  
 map containers, 656  
 memory allocation errors and, 292–293  
 for menu messages, 817–820  
 multidimensional arrays passed to, 253–255  
 need for, 241  
 nested `try` blocks, 290–291  
 overloading, 293–296  
 overview, 239–240, 279  
 parameters, initializing, 285–287  
 pass-by-pointer, 249–251  
 pass-by-reference, 255–257  
 pass-by-value, 247–249  
 pointers, notation when passing arrays, 252–253  
 pointers, returning, 263–264  
 pointers as arguments to, 249–251  
 pointers to, 279–285  
 pointers to, declaring, 280–283  
 priority queue containers, 643  
 prototypes, 243–247  
 queue containers, 640  
 recursive, 271–274  
 references, as arguments to, 255  
 references, returning, 266–268  
 return statements, 243  
 stack containers, 649  
 static variables in, 269–270  
 structure of, 241–243  
 summary, 277  
 templates, 296–299  
 templates in STL, 601

try blocks, 289–291

using, 243–247

values returned from, 263–268

virtual. See *virtual functions*

### functions, C++/CLI, 315–329

arguments to `main()`, 275–276  
 arithmetic expressions, evaluating, 323–325  
 calculator program for CLR, 322–328  
 exercises, 277–278  
 functions, specifying new, 570–571  
 generic, 316–322  
 input strings, removing spaces from, 323  
 numbers, evaluating, 325–326  
 overview, 274–275, 315–316  
 parenthesized substrings, extracting, 327–328  
 summary, 277  
 value of terms, 325

**fundamental data types.** See *data types*

## G

### GDI (Graphical Device Interface)

device context, 849  
 mapping modes, 849–851  
 overview of, 848–849

### generic classes, C++/CLI

collection classes, 590–595  
 interface classes, 589–590  
 overview of, 585–588

### generic functions, C++/CLI, 316–322

defining, 317  
 overview, 316  
 working with, 317–322

`GetDocExtent()`, for determining document size, **1068–1069**

`GetMessage()`, retrieving Windows messages, **759–761**

**global functions, for `CBox` class, 463–465**

**global scope, 89**

**global variables, 89–92**

### Graphical Device Interface (GDI)

device context, 849  
 mapping modes, 849–851  
 overview of, 848–849

### graphical user interface (GUI)

methods for creating interactive Windows applications, 744  
 Windows Forms applications, 1217–1218

**Graphics class**

- coordinate system transformations and, 965–967
- `Draw()`, 899
- `DrawString()`, 1038

**graphics display modes, 852–853****group box, pen width options, 989****GroupBox controls**

- `Controls` property, 1233
- Windows Forms, 831
- working with, 1222–1224

**GUI (graphical user interface)**

- methods for creating interactive Windows applications, 744
- Windows Forms applications, 1217–1218

**H****.h file, DLL file requirements, 1105****handles, integer values for identifying objects, 752****hash values, map keys and, 919–920****header files, 43****headers**

- column headers, for `DataGridView` control, 1280–1281
- `DataGridView` controls, customizing, 1276–1277
- function headers, 241–242
- STL container adapters, 603
- STL containers, 602

**heap vs. free store, 194****Help>About menu items, Windows Forms applications, 1253****helper functions**

- array collections, 913
- list collections, 917–918
- map collections, 919–920

**hexadecimal, ASCII codes, 1309–1315****highlighting elements**

- cell under mouse cursor, 1286–1289
- for selection, 952–956
- for selection in CLR, 972–976

**Hungarian notation**

- MFC and, 771
- Windows notation and, 750

**I****IComparable, interface class, 561****IContainer, interface class, 562–565****IController, interface class, 562****IDE (Integrated Development Environment)**

- compiler in, 9
- components of, 9
- console applications, files created by building, 19
- debugging and releasing program versions, 20
- defined, 9–10
- dockable toolbars, 12–13
- documentation in, 13
- editor in, 9
- error handling, 23–24
- executing program versions, 20–21
- libraries in, 10
- linker in, 9
- MFC applications, building and executing, 30
- MFC applications, creating, 27–29
- options, setting, 26–27
- projects, creating CLR console, 24–26
- projects, creating empty console, 21–23
- projects, creating Win32 console application, 14–19
- projects, defining, 14
- projects and solutions, 13–14
- solutions, building, 19
- source code, modifying, 17–19
- summary, 33–34
- toolbar options, 11–12
- Windows applications, 27–30
- Windows Forms applications, 30–33
- working with, 10–11

**identifiers, 49****if statements, 119–126**

- basic, 119–120
- extended, 122–124
- nested, 120–122, 153–154
- nested `if-else`, 124–126

**IMPLEMENT\_DYNCREATE() macro, 1049–1050****implicit boxing, 554****implicit casts, 69****#include statements, 43, 309, 512****incomplete definitions, classes, 538****incorrect results, common bugs, 698****increment operators (++) , overloading, 436–437****indefinite for loops, 140–141****Indent(), indenting debug output, 734****indexed properties, C++/CLI**

- complex, 401–402
- defined, 390
- defining, 397–398
- working with default, 398–401

**indirect base classes, 509**

## indirect data access

### indirect data access, 174–194

- address-of operators, 175–176
- array names as pointers, 189–191
- arrays of pointers, 181–183
- char, 179–181
- constant pointers and pointers to constants, 185–187
- counting characters, 191–192
- declaring pointers, 175
- defined, 174
- indirection operators, 176
- initializing pointers, 178–179
- multidimensional arrays, pointer notation with, 193–194
- multidimensional arrays, using pointers with, 192–193
- pointer arithmetic, 187–189
- pointers, defined, 174
- pointers and arrays, 187–192
- sizeof operators, 183–185
- using pointers, 176–178

### indirect member access operators, 340

### indirect member selection operators, 340

### indirection operators, 176

### inheritance. *See* class inheritance

### initial values, for variables, 51

### initializing

- arrays, 166–167
- defined, 51
- multidimensional arrays, 172–173
- parameters, 285–287
- pointers, 178–179
- references, 199–200
- structs, 333

### InitInstance(), SDI application, 799–800

### initonly fields, 403–404

### input, analyzing, 300–302

### input, keyboard

- C++/CLI, 106–107
- for calculations, 68–69
- generally, 60

### input controls, disabling, 1251

### input stream iterators, 665–668

### input strings, 323

### input validation, Windows Forms applications, 1242–1244

### input/output operations, 60–64

- escape sequences, 63–64
- input from keyboards, 60

- output, formatting, 61–62
- output to command lines, 60–61

### insert()

- elements in a list, 630–631
- elements in a vector, 616–617
- elements into multimap containers, 664
- pairs into a map, 653–654
- string objects, 478–479

### inserter iterators, 669–672

### insertion operators (<<), CArchive class, 1051

### instances, 340–342

- of class templates, 438

### instantiation, 343

- of class templates, 438

### integer type modifiers, 54–55

### integer variables, 52–53

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

### IntelliSense, 337–338

### interface, DLL, 1098

### interface class keyword, 561

### interface classes, C++/CLI, 561–565

- defining, 561–562
- generic, 589–590
- implementing, 562–565
- overview of, 561

### interface struct keyword, 561

### interfaces

- class interface, 446–447
- DLLs, 1105
- document interfaces, MFC, 780
- GDI (Graphical Device Interface). *See* GDI (Graphical Device Interface)
- GUI (graphical user interface), 744, 1217–1218

### interfaces, C++/CLI

- access specifiers for, 566–568
- visibility specifiers for, 565

### interior pointers, 233–237

### intermediate languages, 2

### internal, C++/CLI access specifier, 566–568

### ISO/ANSI C++

- compiler, 455
- fundamental data types, 56–57
- keywords, 1307–1308
- learning about, 5
- standards, 5–6

### iterators

- begin() and end(), 609
- categories of, 604

input stream iterators, 665–668  
 inserter iterators, 669–670  
 output stream iterators, 670–672  
 overview of, 604–605

**J****joining strings**

C++/CLI, 224–225  
 null-terminated, 201–203

**joins, table**

table joins, 1118–1120  
 table joins in, 1116

**K****key presses, C++/CLI, 154–156****keyboard input**

C++/CLI, 106–107  
 for calculations, 68–69  
 generally, 60

**keyboards, in event-driven programs, 7–8****keys, hashing and, 919–920****keywords**

C++, 1307  
 C++/CLI key, 1308  
 operator keyword, 421  
 variables, 50

**L****late binding (runtime dynamic linking), 1097–1098****length, of null-terminated strings, 200–201****Length bytes, hash values and, 919–920****Length class**

implementing assignment operator, for reference types, 503  
 overloading increment and decrement operators, 499–500  
 overloading operators, in reference classes, 500–502  
 overloading value classes, 494–498

**less than operator, 720–722****.lib files, DLL file requirements, 1105****libraries. See also STL (Standard Template Library)**

functions stored in, 1093  
 MFC (Microsoft Foundation Classes), 10  
 MSDN (Microsoft Development Network) Library, 13  
 .NET library, 833

Standard C++ Library, 10, 43  
 string functions in, 200–206

**Limits menu, event handlers, 1235, 1252–1253****Line class, defining CLR lines, 898****lines**

calculating enclosing rectangle, 880  
 defining for CLR drawings, 898–899  
 drawing, 853–854, 875  
 drawing in CLR drawings, 899–900

**LineTo(), 853–854****list box controls**

adding list to, 1237–1238  
 creating, 1018  
 dialog class for, 1019–1020  
 displaying dialog for, 1020  
 overview of, 1018

**list collections**

adding elements, 913–915  
 deleting objects, 917  
 drawing a curve, 924–925  
 helper functions, 917–918  
 iterating through, 916  
 overview of, 913  
 searching, 916–917  
 using, 923

**list containers, STL/CLR, 630–639**

accessing elements in, 631  
 adding elements, 630–631  
 filtering, 636–639  
 operations performed on, 633–636  
 overview of, 630  
 for storing values types, 684–685  
 working with, 631–633

**ListBox objects, 1239****listeners, setting output destination with, 734****literal fields, 385–386****literals, 57–58****load-time dynamic linking (early binding), DLLs and, 1097****local scope, of automatic variables, 86****logical coordinates**

coordinate systems, 939–941  
 mapping mode and, 1009

**logical operators**

AND, 127  
 combining, 129  
 NOT, 128  
 OR, 128

**logical variables, 55**

## loops

### loops, 135–152

- continue statements, 142–143
- defined, 135
- do-while, 148–149
- for each, 156–159
- floating-point loop counters, 145–146
- for, 136–138
- indefinite for, 140–141
- key presses and, 154–156
- multiple counters, using, 139–140
- nested, 149–152
- for repeating blocks of statements, 135
- types in, other, 143–145
- variations on for, 138
- while, 146–148

### loops, C++/CLI, 153–160

- for each loops, 156–159
- exercises, 159–160
- nested if statements, 153–154
- overview, 153
- summary, 159

lvalues, 65

## M

### main() function

- arguments to, 258–260, 275–276
- defined, 44
- for implementing calculators, 310–311

make\_pair(), helper function for creating pair objects, 653

### managed C++, 3–4

### manipulators, 62

### map collections, 918–920

- helper functions, 919–920
- overview of, 918–919

### map containers, STL/CLR, 652–664

- accessing objects, 654–655
- functions, 656
- implementing phone book using map, 686–692
- operations performed on, 655–656
- overview of, 652
- storing objects, 652–654
- working with, 656–664

### mapping modes

- GDI (Graphical Device Interface), 849–851
- scalable, 1008–1009
- scaling and, 1008
- settings, 1010–1012

### masked elements, 964–965

### MDI (Multiple Document Interface)

- command messages, processing, 811
- creating MDI applications, 801–802
- overview of, 780

### member access operators, 333

### member functions

- class templates, 440–441
- definitions, 369–371

### member selection operators, 333

### members

- of classes, 343
- defined, 332

### memory, debugging dynamic

- free store debug operations, checking, 723–724
- free store debug operations, controlling, 725–726
- free store debug operations, output, 726–727
- leak detection, 727–731
- overview of, 723

### memory, sharing between variables, 417

### memory allocation errors, 292–293

### memory leaks

- causes of, 723
- detecting, 724, 727–731
- difficulty in detecting, 723

### menu bar

- adding menu items to, 813
- Color menu added to, 815
- as window element, 746

### menu items

- adding to Element menu, 814–815
- adding to menu bar, 813
- adding to spin button control, 1002–1003
- Choose menu item, 1257
- in context menus, 950–952
- event handlers for, 1229–1230
- Help>About, 1253
- Limits menu, 1235, 1252–1253
- Menu Strip container for, 836–837
- modifying existing, 815
- Reset, 1245
- text, adding, 1025–1026

### menu items, CLR

- checks, 840–842
- event handlers, 838–839

### menu messages

- classes, 817
- coding functions, 820
- creating functions, 817–820
- deleting elements, 957–958
- event handlers, 816

## MouseMove event handler

- exercising, 963
- getting elements to move themselves, 961–963
- modifying `WM_MOUSEMOVE` handler, 959–961
- moving elements, 958–959
- servicing, 956
- Menu Strip container, for menu item, 836–837**
- menus**
  - adding to programs, 811
  - completing the menu for Sketcher application, 815
  - creating and editing menu resources, 812
  - elements of, 812
  - submenus, 1219–1220
- menus, CLR, 836–838**
- Menus & Toolbars **group**, **Toolbox window, 1228**
- `merge()`, **deleting elements from a list, 635–636**
- message handlers**
  - class definitions and, 807–809
  - coding update handler, 825–827
  - exercising update handler, 827
  - mouse, 865–867, 885
  - radio buttons, 997–998
  - for updating user interface, 824–825
- message loop, retrieving Windows messages, 758–759**
- message maps, established by MFC Application Wizard, 806–807**
- message pump, retrieving Windows messages, 758–759**
- message queues, Windows messages, 747**
- `MessageBox` **class, 1253**
- messages**
  - categories of, 809–810
  - message type relationship to menu items, 816
  - MFC handling, 810
  - from mouse, 864–865
- messages, Windows. See Windows messages**
- methods, 343**
- MFC (Microsoft Foundation Classes), 779–804**
  - application class (`CWinApp`), 772
  - Application Wizard, location of project files, 790–791
  - applications, creating, 27–29
  - applications, executing, 30
  - basic classes, for Windows applications, 783
  - class definitions, 793–797
  - classes, viewing, 792–793
  - database applications. *See* database applications
  - defined, 3–4
  - DLLs and, 1095
  - document templates, 782–783
  - documents, 780
  - document/view concept in, 780
  - executable modules, 797–801
  - exercise creating MFC application, 771–775
  - exercises, 803–804
  - extension DLLs. *See* extension DLLs
  - as library, 10
  - linking documents and views, 781–782
  - MDI applications, 801–802
  - notation, 771
  - `OnDraw()` member function, 851–852
  - overview of, 770–771, 779–780
  - project files, viewing, 792
  - SDI applications, 786–790
  - structure of MFC programs, 771
  - summary, 803
  - views, 780–781
  - window class (`CFrameWnd`), 773
  - Windows applications, creating, 785–786
  - Windows applications, interactive, 744
  - Windows applications, tools for creating, 784–785
  - in Windows OSs, 10
- Microsoft Development Network (MSDN)**
  - library, 13
  - updates and fixes from, 697
- Microsoft Foundation Classes. See MFC (Microsoft Foundation Classes)**
- Microsoft Intermediate Language (MSIL), 2**
- `MM_ANISOTROPIC`, **1008–1009, 1068**
- `MM_ISOTROPIC`, **1008–1009**
- `MM_LONGENGLISH`
  - document views, 941–943
  - mapping modes, 850–851
  - multipage printing and, 1068
- `MM_TEXT`, **as default mapping mode, 850**
- modal dialog boxes, 992**
- modeless dialog boxes, 992**
- mouse**
  - drawing with, 867–869
  - event handlers for CLR drawings, 895–897
  - in event-driven programs, 7–8
  - highlighting cell under cursor, 1286–1289
  - message handlers, 865–867, 885
  - messages, capturing, 893–895
  - messages from, 864–865
  - programming, 863–864
  - properties, 865
  - using for graphics, 861–862
- `MouseMove` **event handler, CLR drawings, 904–905**

## MouseUp event handler

MouseUp **event handler, CLR drawings, 905**

Move(), **getting elements to move themselves, 961**

MoveTo(), **displaying graphics and, 853**

**MSDN (Microsoft Development Network)**

library, 13

updates and fixes from, 697

**MSIL (Microsoft Intermediate Language), 2**

**multidimensional arrays**

in C++/CLI, 217–221

defined, 171–172

dynamic allocation of, 198–199

initializing, 172–173

passing to functions, 253–255

pointer notation with, 193–194

pointers with, 192–193

**multifile project, CBox class, 453–455**

**multimap containers, 664–665**

**multipage printing, 1066–1068**

**multiple counters, 139–140**

**multiple data values, of same type, 162–174**

arrays, declaring, 163–166

arrays, defined, 162–163

arrays, initializing, 166–167

character arrays and string handling, 168–169

multidimensional arrays, defined, 171–172

multidimensional arrays, initializing, 172–173

multiple strings, storing, 173–174

overview, 162

programming with strings, 170–171

string input, 169–171

**Multiple Document Interface (MDI)**

command messages, processing, 811

creating MDI applications, 801–802

overview of, 780

**multiple namespaces, 96–97**

**multiple parameters, class templates, 444–446**

**multiple strings, 173–174**

**multiple top-level document architecture, MFC**

**document interfaces, 780**

**multiset containers, 651**

**multitasking, Windows programs, 759**

multop number, **301**

## N

**named indexed properties, 397**

**names**

functions, 241

as pointers, 189–191

program files, 470–471

variables, 49

**namespaces**

declaration and, 43–44

declaring, 94–96

defined, 25–26

multiple, 96–97

.NET library, 833

overview, 93–94

structure of C++ programs, 43–44

**native C++ programs, defined, 2–5**

**native enumerations, 112**

**NDEBUG preprocessor symbol, 708**

**nested classes**

overview of, 550–552

working with, 552–553

**nested if statements, 120–122**

**nested if statements, C++/CL, 153–154**

**nested if-else statements, 124–126**

**nested loops, 149–152**

**nested try blocks, 290–291**

**.NET Framework, 2, 1078**

**.NET library namespaces, 833**

**new keyword, 570–571**

**new operators**

constructors and, 412

dynamic memory allocation and, 195

**noarg constructors, 353**

**non-queued messages, Windows programs, 758**

**normalized rectangles, CLine class, 880**

**NOT operators, 84, 128**

**notation**

MFC (Microsoft Foundation Classes), 771

pointers, 193–194

Windows programming, 749–750

**null pointer, 705**

**null-terminated strings**

comparing, 204

copying, 203–204

joining, 201–203

length of, 200–201

searching, 204–206

**number() function, calculators, 306–309, 311–312**

**numbers, evaluating, 325–326**

## O

**object code, 9**

**object files, 9**

**object-oriented programming.** See **OOP (object-oriented programming)**

**objects**

- accessing in map containers, 654–655
- CBox objects, analyzing, 451–452
- CBox objects, combining, 449–451
- CBox objects, comparing, 448–449
- CLR serialization, 1081–1082
- created from class templates, 441–444
- creating string objects, 471–473
  - as data source, 1266
- data types, 340–342
- overview of, 343
- storing in map containers, 652–654

**objects, collections of**

- CArray template class, 911–912
- CList template class, 913–918
- CMap template class, 918–920
- overview of, 911

**.ocx (OLE Custom Extension), 1095**

**ODBC (Open Data Base Connectivity)**

- database application. See database applications
- drivers, 1121
- generating ODBC program, 1124–1125
- MFC classes supporting, 1121–1122
- registering ODBC databases, 1122–1124
- updating data sources. See databases, update operations

**old-style casting, 79**

**OLE Custom Extension (.ocx), 1095**

**OLE DB, database support in MFC, 1120**

OnContextMenu(), **946–947**

OnDraw(), **MFC, 851–852, 931–933**

**one-dimensional arrays, C++/CLI**

- searching, 215–217
- sorting, 213–214

OnInitDialog(), **996–997, 1006**

OnMouseMove()

- coding mouse move message handler, 887–889
- identifying selected element, 948–949
- message handler, 885

OnPreparePrinting(), **1069**

OnPrint(), **printing documents, 1073–1077**

**OOP (object-oriented programming)**

- basic concepts, 507–509
- overview, 341–343
- polymorphism and, 535

**open, file operations, 1087–1088**

**Open Data Base Connectivity.** See **ODBC (Open Data Base Connectivity)**

**operations, checking legality of database update operations, 1169**

operator()(), **421, 605–606**

operator keyword, **421**

**operator overloading**

- addition operators, 432–436
- assignment operators, 427–431
- comparison operators, 424–426
- implementing, 420–423
- increment and decrement operators, 436–437
- overview of, 420

**operator overloading, C++/CLI**

- decrement operators, 499–500
- reference classes, 500–502
- value classes, 494–498

operator<(), **operator overloading, 424–426**

operator>(), **operator overloading, 422–423**

**operators**

- addition operators, overloading, 432–436
- analytical operators, 451–452
- arithmetic operators, 449–451
- assignment operators, overloading, 427–431
- comparison operators, 448–449
- comparison operators, overloading, 424–426
- increment and decrement operators, overloading, 436–437
- overloading, 420–423
- precedence of, 74–76
- strings, 1024–1025

**operators, C++/CLI**

- decrement operators, overloading, 499–500
- reference classes, 500–502
- value classes, 494–498

**optimistic mode, record locking, 1170**

**options, setting, 26–27**

OR operators, **82–83, 128**

ORDER BY, **sorting recordsets, 1138–1139**

**Order Details table, customizing, 1173–1174**

**order dialog, linking to customer resource dialog, 1163–1165**

**Order IDs**

- creating new, 1201–1204
- initiating ID creation, 1204–1205
- overview of, 1200
- storing, 1201

**orders/ordering, in database application**

- adding new, 1209–1214
- customizing Orders Details table, 1173–1174
- entry process, 1186–1187
- order data, storing, 1205–1207

## orders/ordering (*continued*)

### orders/ordering (*continued*)

- Order IDs, creating, 1200–1205
- products, selecting for orders, 1207–1209

### ordinal values, DLLs, 1104

### output

- to command lines, 60–61, 102
- conditional operators with, 130–131
- formatting, 61–62, 103–106
- free store debug operations, 726–727
- statements, 45

### output, C++/CLI debugging

- controlling, 735–736
- destination, 734
- generating, 732–733
- indenting, 734

### output stream iterators, 670–672

### overloading functions, 293–296

### overriding operator precedence, 76

## P

### packaging problems, class interfaces and, 446–447

### page coordinates. *See* logical coordinates

### Paint event handler

- CLR drawings, 905–906
- drawing sketch with, 971–972

### panes, status bars, 1013

### parameters

- class templates with multiple parameters, 444–446
- defined, 240–242
- initializing, 285–287
- insert(), 478–479
- parameter values, for constructor arguments, 356–357
- parameter values, in classes, 355–356

### parameters, SQL filters

- defining, 1148–1149
- overview of, 1147–1148

### parent windows, window elements, 745

### parenthesized substrings, extracting, 327–328

### parts, status bars, 1015

### pass-by-pointer, 249–251

### pass-by-reference, 255–257

### pass-by-value, 247–249

### passing

- arguments to functions, 247–251
- arrays to functions, 251–253
- multidimensional arrays to functions, 253–255

### pdo\_it function, 282–283

### Pen class, 900–902

### Pen width dialog box

- exercising, 1001
- pen width, adding to document, 998–999
- pen width, adding to elements, 999–1000

### pens

- creating, 857–858
- defining for CLR drawings, 900–902
- pen width options, 989
- setting drawing width, 1033–1035
- standard for CLR drawings, 902
- using, 858–859

### pessimistic mode, record locking, 1170

### pfun() statement, 280–281

### Play menu event, 1231–1235

### playMenuItem\_Click(), 1231

### pointer notation when passing arrays, 252–253

### pointers

- for accessing members of structs, 339–340
- address-of operators, 175–176
- as arguments to functions, 249–251
- arithmetic, 187–189
- array names as, 189–191
- and arrays, 187–192
- arrays of, 181–183
- causes of bad pointers, 698
- char, 179–181
- to class objects, 376–379, 535–537
- constants and, 185–187
- counting characters, 191–192
- declaring, 175
- declaring to functions, 280–283
- defined, 174
- dynamic\_cast operator and, 550
- to functions, 279–285
- indirect data access and, 174
- indirect member selection operators, 340
- indirection operators, 176
- initializing, 178–179
- multidimensional arrays and, 192–193
- notation, when passing arrays, 252–253
- notation, with multidimensional arrays, 193–194
- null pointer, 705
- returning, 263–264
- sizeof operators, 183–185
- storing in vector containers, 624–626
- with structs, 338–340
- working with, 176–178

### polymorphism, virtual functions and, 535

### pop\_back(), deleting elements in a vector, 616

- pop-up menus**
  - context menus, 943
  - exercising, 949–950
  - flags, 946
  - Popup property and, 813
- Popup **property, 813**
- POSITION **values, list collections, 915–916**
- precompiled headers, compilation time and, 797–798**
- predicates**
  - for filtering a list, 636–639
  - overview of, 606
- pre-emptive multitasking, Windows programs, 759**
- prefixes**
  - Windows messages, 1317–1318
  - Windows notation, 749–750
- preprocessor directives, adding debugging code with, 709**
- primary keys, in relational databases, 1114**
- print, file operations, 1089–1090**
- Print(), **generating debugging output with, 733**
- print preview, for documents, 1077**
- PrintDocument **component, 1089**
- printing documents**
  - cleaning up after, 1072
  - CPrintInfo class, 1066–1068
  - document size and, 1068–1069
  - exercises, 1091–1092
  - logic and functions involved in printing process, 1065–1066
  - multipage, 1066–1068
  - overview of, 1064–1065
  - preparing device context for, 1072–1073
  - preparing for, 1070–1072
  - print preview, 1077
  - process of, 1073–1077
  - storing print data, 1069–1070
  - summary, 1090–1091
- printing documents, CLR**
  - overview of, 1089
  - print operation, 1089–1090
  - PrintDocument component, 1089
- priority queue containers, 643–648**
  - functions, 643
  - overview of, 643–644
  - working with, 644–648
- private classes, C++/CL, 565**
- private **keyword, class membership and, 513**
- private members, of classes, 358–366**
  - access levels for inherited members, 522–523
  - accessing, 361
  - accessing private members of base class in a derived class, 513–516
  - default copy constructors, 364–366
  - friend functions and, 362–364
  - overview, 358–359
  - working with, 359–361
- private protected **keyword, C++/CLI access specifiers, 566–568**
- Product ID, in relational databases, 1114**
- products**
  - selecting for orders, 1207–1209
  - viewing product orders, 1151–1155
- Products **table**
  - querying, 1129
  - in relational databases, 1114
- program code**
  - file names, 470–471
  - organizing, 468
- programming mouse, 863–864**
- programs**
  - bugs, 696–697
  - comments, 42
  - hangs/crashes, 698
  - named indexed properties, 397
  - statements, 44–46
  - structure. *See* structures
  - Windows. *See also* Windows programming
- programs, debugging**
  - call stack, 715–716
  - finding next bug, 722
  - overview of, 715
  - program versions, 20
  - stepping over to errors, 716–720
  - testing extended class, 720–722
- project files**
  - location of, 790–791
  - viewing, 792
- project folders, 13**
- projects**
  - Ctrl+Shift+N (New Project), 785, 1100
  - defining, 14
  - solutions and, 13–14
- properties**
  - C++/CLI class properties. *See* class properties, C++/CLI
  - cell styles, 1275–1276
  - columns, 1281–1283
  - controls, 1233
  - defined, 390

## properties (continued)

### properties (continued)

- modifying Windows Forms application, 834–835
- mouse, 865
- read-only, 390
- reserved property names, 403
- toolbar buttons, 829–830
- write-only, 390

### Property Manager tabs, 17

**protected** keyword, class membership and, 519–522

### protected members

- access levels for inherited class members, 522–523
- creating elements and, 889

### prototypes, function, 243–247

### public classes, specifying in C++/CLI, 565

### public data, accessing, 783

**public** keyword, class membership and, 513

### public members

- access control and, 513
- access levels for inherited class members, 522–523
- interface classes, 561

**public protected** keyword, C++/CLI access specifiers, 566–568

### pure virtual functions

- abstract classes, 539–542
- overview of, 538–539

`push_back()`

- adding elements to lists, 630
- adding elements to vector containers, 608

`push_front()`, adding elements to lists, 630

## Q

**queries, 1129–1131**

### queue containers, 640–648

- functions, 640
- priority queue containers, 643–648
- working with, 640–643

**queued messages, Windows programs, 758**

## R

**R2\_NOTXORPEN, 887, 1062**

### radio buttons

- adding to group box, 989–990
- initializing, 996–997
- message handling, 997–998

**Raster OPeration (ROP), 886**

`rbegin()`, reverse iterators, 610

**RDBMS (relational database management systems), 1115**

**read-only properties, 390**

**Record Field Exchange, RFX\_(), 1131–1132**

**record locking, 1169–1170**

**Record menu, disabling, 1181–1183**

### record view class

- creating, 1144–1147
- initializing, 1149–1150
- overview of, 1132–1134

### records

- choosing SQL records, 1117
- in relational databases, 1114
- sorting SQL records, 1120

### recordsets

- adding filters to, 1147–1148
- creating, 1127–1129, 1188
- customer recordset, adding, 1156–1157
- customer recordset, filtering, 1160–1163
- data transfer between database and recordset, 1131–1132
- linking controls to, 1136–1138
- recordset class, 1141–1143
- second recordset object, 1140–1141
- SELECT statements resulting in, 1116–1117
- snapshot vs. dynaset records, 1126
- sorting, 1138–1139
- views, for database update, 1188–1193

`Rectangle()`, **CRectangle** class, 881–882

### rectangles

- calculating enclosing rectangle for lines, 880
- defining for CLR drawings, 902–903
- drawing, 881–882
- RECT structure, 338

**recursive function calls, 271–274**

### reference classes, C++/CLI

- base class for, 553
- copy constructors, 389–390
- defining, 386–387
- derived classes and, 558–560
- destructors and finalizers, 583–585
- implementing assignment operator for, 503
- overloading, 500–502
- storing reference class objects, 680–684
- working with, 387–389

### references

- as arguments to functions, 255
- to base classes, 537
- to class objects, 376, 379–380

- declaring and initializing, 199–200
  - defined, 199
  - dynamic\_cast operator and, 550
  - returning, 266–268
  - using, 199–200
  - to virtual functions, 537–539
  - references, C++/CLI, 555**
  - RegisterClassEx(), **creating program window, 756–757**
  - registering window class, 756**
  - Regular DLL, dynamically linked to MFC, 1099**
  - Regular DLL, statically linked to MFC, 1099**
  - relational database management systems (RDBMS), 1115**
  - relational databases. See also database applications**
    - components of, 1114
    - as data source, 1266
    - overview of, 1113
    - RDBMS (relational database management systems), 1115–1116
  - relational operators, 117**
  - releases, program versions, 20**
  - remainders, 70–71**
  - remove(), **deleting elements from a list, 634**
  - remove\_if(), **deleting elements from a list, 636**
  - rend(), **reverse iterators, 610**
  - repeating blocks of statements. See loops**
  - replace()
    - algorithms, 674
    - string objects, 479–480
  - reserve(), **increasing container capacity, 609**
  - reserved property names, 403**
  - Reset **menu item, 1245**
  - resize(), **vector containers, 612**
  - Resource Editors, tools for creating MFC-based Windows programs, 785**
  - resource files, as window element, 746**
  - Resource View**
    - adding Tooltips, 831
    - creating view dialogs, 1134–1136, 1143–1144
    - customer resource dialog, 1157–1158
    - customer resource dialog, linking to order dialog, 1163–1165
    - extending toolbar resources, 827–828
    - IDE windows, 10–11
    - overview of, 17
    - switching from Solution Explorer to, 812
  - resources**
    - adding buttons to dialog boxes, 1187–1188
    - adding controls to dialog resources, 1193–1196
    - creating and editing menu resources, 812
    - extending toolbar resources, 827–828
  - results, displaying, 69–70**
  - return statements, 243**
  - return values, 241**
  - reverse iterators, 610**
  - RFX\_() (**Record Field Exchange**), **1131–1132**
  - roll back, of transactions, 1170**
  - Rollback(), CDatabase **transaction operations, 1171–1172**
  - ROP (Raster Operation), 886**
  - rows**
    - adding to tables, 1186–1187
    - customizing, 1283–1284
    - DataGridView functions for, 1269–1270
    - System::Data::DataRows namespace, 1266
  - rules**
    - for casting operands, 76–77
    - for returning addresses, 265–266
    - for returning references, 268
  - Run(), **SDI application, 800–801**
  - Run to Cursor option, Debug menu, 704**
  - runtime dynamic linking (late binding), 1097–1098**
  - rvalues, 65
- ## S
- safe\_cast, 107
  - save, file operations, 1086–1087**
  - save as, file operations, 1088**
  - scalable mapping modes, 1008–1009**
  - scalar properties, C++/CLI**
    - defining, 390–393
    - trivial, 393
    - working with, 393–397
  - scale dialog**
    - displaying for spin button control, 1002
    - removing, 1018
    - scale dialog class, 1004–1005
  - scale factor**
    - document size settings, 1009–1010
    - mapping mode settings, 1010–1012
    - overview of, 1008
    - scalable mapping modes and, 1008–1009
    - scrolling with scaling, 1012–1013
  - scale menu item, adding to spin button control, 1002–1003**
  - scope resolution operators**
    - in classes, 349
    - defined, 43
    - storage duration and scope, 91–92

## screen coordinates

### screen coordinates, pixels for measuring, 1009

#### scrolling

- scaling and, 1012
- scroll bar controls, 1012–1013
- views, 937–939

#### SDI (Single Document Interface)

- Build Solution, 797
- class definitions, 793–797
- command messages, 810
- creating SDI applications, 786–790
- executable modules, 797–801
- location of project files, 790–791
- overview of, 780
- precompiled headers and, 797–798
- viewing classes, 792–793
- viewing project files, 792

#### searches

- Find(), searching list collections, 916–917
- FindIndex(), searching list collections, 917
- null-terminated strings, 204–206
- one-dimensional arrays, 215–217
- strings, 224–225, 230–233, 484–489

#### SELECT statements

- filters, 1147
- in relational database, 1116
- snapshot vs. dynaset records, 1126
- sorting records, 1120
- table joins, 1118
- WHERE clause, 1117

#### SelectObject(), 858

#### SelectView(), 1151–1152

#### semantic errors, 697

#### Send-To-Back operation, in CLR, 978

#### sequence, of calculations, 74–76

#### sequence containers, 607–651

- basic types, 607
- defined, 606
- double-ended queue containers, 626–629
- list containers, 630–639
- overview of, 607–608
- queue containers, 640–648
- stack containers, 649–651
- vector containers. See vector containers

#### sequence containers, STL/CLR

- double-ended queue containers for storing reference class objects, 680–684
- list containers for storing values types, 684–685
- overview of, 677
- vectors for storing handles, 678–680

#### Serializable attribute, 1083

#### serialization

- CArchive class, 1050–1052
- CObject class, 1052
- collections and, 910
- DECLARE\_SERIAL() macro, 1052–1053
- in document class definition, 1048–1049
- in document class implementation, 1049–1050
- exercises, 1091–1092
- exercising, 1061–1062
- how it works, 1053–1054
- implementing for a class, 1054–1055
- overview of, 1047–1048
- recording document changes, 1055
- Serialize(), for documents, 1056–1057
- Serialize(), for elements, 1058–1059
- Serialize(), for shapes, 1059–1061
- Serialize(), overview of, 1050
- summary, 1090–1091

#### serialization, CLR

- binary serialization, 1078–1079
- fields that are not serializable, 1079–1081
- file operations, for a sketch, 1086–1088
- object serialization, 1081–1082
- overview of, 1078
- sketch serialization, 1083–1086

#### Serialize()

- definition of, 1050
- document objects and, 1049
- how it works, 1053–1054
- implementing for documents, 1056–1057
- implementing for elements, 1058–1059
- implementing for shapes, 1059–1061

#### set containers, 651

#### SetCapture(), capturing mouse messages, 893–895

#### SetCheck(), CCmdUI class, 826

#### SetROP2(), 886

#### SetTextAlign(), 1077

#### SetValues(), 1233–1234

#### shapes

- of collections, 910
- deleting and moving, 943
- serialization, 1059–1061

#### shift operators, 84–86

#### Shift+Alt+D (Show Data Sources item), 1266

#### Show(), 1242–1243

#### Show Data Sources item (Shift+Alt+D), 1266

#### ShowDialog(), 1240–1241

#### ShowWindow(), 757–758

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

## STL (Standard Template Library)

- `size()`, **vector containers**, 611
- `sizeof` operators, 183–185
- Sketcher application**
  - CLR version. See Windows applications, CLR-based
  - MFC version. See Windows applications, MFC-based
- sketches, CLR serialization**, 1083–1086
- snapshot, vs. dynaset records**, 1126
- SOAP serialization**, 1078
- Socket class, in STL**, 601
- Solution Explorer**
  - IDE windows, 10–11
  - overview of, 16
  - switching to Resource View, 812
- solutions, 13–14, 19**
- `sort()`, 623–624
- sorting elements**, 623–624
- sorting recordsets**, 1138–1139
- sorting SQL records**, 1120
- sorting vector elements**, 623–624
- sorting words from text**, 489–494
- source code, modifying**, 17–19
- spin button control**
  - creating, 1003–1004
  - defined, 988
  - dialog data exchange and validation, 1006
  - displaying, 1007–1008
  - initializing the dialog, 1006–1007
  - overview of, 1001–1002
  - scale dialog class, 1004–1005
  - scale menu item, 1002–1003
  - tab sequence, 1004
  - toolbar buttons, 1002–1003
- `splice()`, **deleting elements from a list**, 635
- SQL (Structured Query Language)**
  - overview of, 1116
  - records, 1117
  - retrieving data with, 1116–1117
  - SELECT statements. See SELECT statements
  - sorting records, 1120
  - table joins, 1118–1120
- stack containers**
  - functions, 649
  - STL (Standard Template Library), 649–651
- stacks, in automatic variables**, 86–87
- Standard C++ Library**, 10, 43
- standard libraries**, 1093. See also libraries
- Standard Template Library. See STL (Standard Template Library)**
- standards, learning**, 5–6
- Start Debugging, Debug menu**, 703, 722
- state**
  - of free store, 724
  - of `Listbox` objects, 1239
- statement blocks**, 47
- statements, 44–46**
  - `#include`, 43, 309, 512
  - assignment, 45, 65, 77–78
  - extended `if` statements, 122–124
  - `if` statements. See `if` statements
  - loops. See loops
  - nested `if` statements, 120–122
  - nested `if` statements, C++/CL, 153–154
  - nested `if-else` statements, 124–126
  - `pfun()` statement, 280–281
  - SELECT statements. See SELECT statements
  - `throw` statement, 289–291
- static constructors**, 405
- static members, of classes**, 373–376
  - data members, 373–374
  - function members, 376
  - instances, counting, 374–375
- static properties**, 402–403
- static storage duration**, 89–93
- static variables**
  - in functions, 269–270
  - storage duration and scope, 93
- statically linked functions, DLLs compared with**, 1093–1095
- status bar controls**
  - adding to frames, 1014–1015
  - overview of, 1013–1014
  - panes, 1013
  - parts, 1015–1017
  - updating, 1017–1018
- std namespace**, 601
- Step Into option, Debug menu**, 704
- Step Over (F10), Debug menu**, 704–705, 705, 716–720
- STL (Standard Template Library), 601–693**
  - algorithms, 605, 674–676
  - associative containers. See associative containers
  - container adapters, 603
  - containers, 602–603
  - double-ended queue containers, 626–629
  - exercises, 693
  - function adapters, 606
  - function objects, 605–606, 672–673
  - input stream iterators, 665–668
  - inserter iterators, 669–670
  - iterators, 604–605

## STL (Standard Template Library) (continued)

### STL (Standard Template Library) (continued)

list containers. See list containers, STL/CLR  
 map containers. See map containers, STL/CLR  
 multimap containers, 664–665  
 output stream iterators, 670–672  
 overview of, 601–602  
 queue containers. See queue containers  
 sequence containers. See sequence containers  
 stack containers, 649–651  
 summary, 692–693  
 vector containers. See vector containers

### STL/CLR, 676–692

associative containers, 685–692  
 containers, 677  
 overview of, 676–677  
 sequence containers. See sequence containers, STL/CLR

### storage

class objects, in vector containers, 618–622  
 handles, in vector containers, 678–680  
 multiple strings, 173–174  
 objects, in map containers, 652–654  
 order data, 1205–1207  
 Order IDs, 1201  
 pointers, in vector containers, 624–626  
 print data, 1069–1070  
 reference class objects, 680–684  
 temporary elements, 873–874  
 values types, in list containers, 684–685

### storage, duration and scope, 86–93

automatic variables in, 89  
 declarations, positioning variable, 89  
 global variables, 89–92  
 for multiple strings, 173–174  
 overview, 86  
 scope resolution operators, 91–92  
 static variables, 93

`strcat()`, **debugging programs and**, 715–716

`strcpy()`, 413

`strcpy()`, compared with `strcpy_s()`, 714

`strcpy_s()`

compared with `strcpy()`, 714  
 as derived class, 511

### stream input data incorrect, common bugs, 698

### stream iterators

input, 665–668  
 output, 670–672

### strings, 200–206

accessing characters in, 156–160  
 accessing/modifying, 476–480  
 comparing, 480–484

comparing null-terminated strings, 204  
 concatenating, 473–476  
 copying null-terminated strings, 203–204  
 creating string objects, 471–473  
 handling, 168–169  
 input, 169–171  
 joining null-terminated strings, 201–203  
 length of null-terminated strings, 200–201  
 overloaded operators, 1024–1025  
 overview, 200, 471  
 programming with, 170–171  
 searching, 484–489  
 searching null-terminated strings, 204–206  
 sorting words from text, 489–494

### strings, C++/CLI, 224–233

comparing, 229  
 joining, 224–225  
 modifying, 227–229  
 overview, 224  
 searching, 230–233  
 working with, 225–227

`strlen()`, 413

### Stroustrup, Bjarne, 342

structs, 332–340

accessing members of, 333  
 defining, 332–333  
 indirect member selection operators, 340  
 initializing, 333  
 pointers with, 338–340  
 RECT, 338  
 working with, 334–337

**structure, C++/CLI. See functions, C++/CLI**

**Structured Query Language. See SQL (Structured Query Language)**

**structures, 42–48. See also functions**

automatically generated console programs, 47–48  
 comments, 42  
 declarations, 43–44  
 header files, 43  
 main() function, 44  
 MFC programs, 771  
 namespaces, 43–44  
 #include directive, 43  
 overview, 36–42  
 statement blocks, 47  
 statements, 44–46  
 unions in, 419  
 whitespace, 46–47  
 Windows programs as, 750–751

**submenus, Windows Forms applications, 1219–1220**

**sumarray() function, 284–285**  
**swap(), swapping contents of two vectors, 618**  
**switch statements, 131–134**  
**symbols**  
   exporting from DLLs, 1109–1111  
   importing into a program, 1109–1110  
**synonyms, for data types, 58**  
**syntactic errors, 697**  
**System, 833**  
**system menu, window elements, 745**  
**System::Collections, 833**  
**System::Collections::Generic, 590**  
**System::ComponentModel, 833**  
**System::Data, 833, 1266**  
**System::Data::DataColumn, 1266**  
**System::Data::DataRows, 1266**  
**System::Data::DataTable, 1266**  
**System::Diagnostics, 732**  
**System::Drawing, 833**  
**System::Drawing, 895**  
**System::Drawing::Brush, 1039**  
**System::Drawing::Color, 839, 899**  
**System::Drawing::Font, 1039–1040**  
**System::Drawing::FontStyle, 1039**  
**System::Drawing::Graphics, 899**  
**System::Drawing::Pens, 900–902**  
**System::Drawing::Rectangle, 903**  
**System::IO::Stream, 1081**  
**System::Object, 553, 554, 555**  
**System::Runtime::Serialization, 1079**  
**System::Runtime::Serialization::Formatters::Binary, 1082, 1086**  
**System::Windows::Forms, 833, 1267**  
**System::Windows::Forms::Form class, 775**

## T

**tab controls, adding to Windows Forms application, 1220–1222**  
**tab sequence controls, 1004**  
**TabControl controls**  
   adding, 1220–1221  
   working with multiple tables, 1302–1304  
**table adapters, 1266**  
**table joins**  
   in relational database application, 1116  
   SQL, 1118–1120  
**table views**  
   accessing multiple views, 1150–1151  
   switching, 1151–1154

**tables**  
   adding rows, 1186–1187  
   customizing Orders Details table, 1173–1174  
   in relational databases, 1114  
   System::Data::DataTable namespace, 1266  
   working with multiple, 1302–1304  
**template classes. See also STL (Standard Template Library)**  
   CArray, 911–912  
   CList, 913–918  
   CMap, 918–919  
   CTypedPtrList, 920–923  
   overview of, 911  
**template libraries, C++/CLI. See STL/CLR**  
**templates, function, 296–299**  
**term() function, 305–306**  
**ternary operators, 130**  
**testing, dialog boxes, 988–990**  
**text**  
   base and derived classes, 1026  
   constructor for CText class, 1027–1028  
   drawing text, 1028  
   drawing text, in CLR, 1038  
   implementing, 1027  
   moving text objects, 1028, 1062–1064  
   strings sorting words from text, 489–494  
   text dialog, in CLR, 1042  
   text elements, creating, 1028–1029  
   text elements, for CLR controls, 1037  
   text elements, for CLR text boxes, 1043  
   text elements, for edit boxes, 1026–1027  
**text alignment, printing documents and, 1076**  
**text box controls**  
   overview of, 1042–1043  
   Windows Forms and, 831  
**Text Editor**  
   creating MFC-based Windows programs, 785–786  
   viewing variables in, 707  
**text elements**  
   for CLR controls, 1037  
   for CLR text boxes, 1043  
   creating, 1028–1029  
   for edit boxes, 1026–1027  
**text menu item, adding to edit controls, 1025–1026**  
**TextElement class**  
   creating, 1043  
   defining, 1041  
**this pointer, 366–368**  
**throw statement, 289–291**  
**title bar icon, window elements, 745**

## title bars

### title bars, window elements, 745

to `main()` function, 258–260, 275–276

### toolbar buttons

- adding, 827–828, 992
- editing properties, 829–830
- exercising, 830
- for spin button control, 1002–1003

### toolbars

- options, 11–12
- as window element, 746

### toolbars, CLR, 842–844

### Toolbox window

- displaying, 1220
- Menus & Toolbars group, 1228
- standard dialog boxes, 1235–1236
- View menu, 836

### ToolStrip controls, 842

### tooltips, 830

`Tostring()`, 384–385, 553

### trace

- outputting trace information, 741
- switching debugging/tracing on/off, 735

### Trace classes, C++/CLI

- overview of, 732
- working with, 737–741

### tracepoints, 702–703

`TraceSwitch`, 735–736

### tracking handles, 207–209

### tracking references, 233

### transactions

- `CDatabase` class, 1171–1172
- overview of, 1170

`transform()`, algorithms, 675–676

### transformations, coordinate system, 965–967

`TranslateMessage()`, retrieving Windows messages, 759–760

### tree controls, 988

`try` blocks, 290–291

### type modifiers

- const, 67–68
- defined, 54–55

### type safe collections

- `CArray` template class, 911–912
- `CList` template class, 913–918
- `CMap` template class, 918–920
- overview of, 910–911

### typed pointer collections

- `CTypedPtrList` template class, 921–923, 929–931
- overview of, 920

### typed pointer list collections, 921–923

- creating sketch document with, 929–931
- functions, 921–923
- overview of, 921

`typeid()`, determining class type with, 1063

types. *See* data types

## U

### unary predicates, 606

unbound mode, `DataGridView` controls, *see* `DataGridView` controls, in unbound mode

### unconditional branching, 134

### unhandled exceptions, 698

### Unicode codes, 1315

### unions

- anonymous unions, 419
- in classes and structures, 419
- between variables, 417–419

`unique()`, deleting elements from a list, 634

### unmanaged C++, 3–5

`Update()`, `CRecordset` update operations, 1168–1169

### update handlers

- coding command update handler, 825–827
- exercising, 827
- updating user interface, 824–825

### update messages, 816–817

### update mode, 1177–1178

`UPDATE_COMMAND_UI`. *See* update handlers

`UpdateAllViews()`, 935–937

updating database operation. *See* databases, update operations

### user interfaces. *See also* interfaces

- class interface, 446–447
- GDI (Graphical Device Interface). *See* GDI (Graphical Device Interface)
- GUI (graphical user interface), 744, 1217–1218
- message handlers for updating, 824–825

## V

### validation

- input validation for Windows Forms, 1242–1244
- spin button control, 1006

### value classes

- base class for, 553
- types of, 98

### value classes, C++/CLI

- defining, 381–382

- overloading, 494–498
- working with, 383–384
- value of terms C++/CLI, 325**
- values**
  - enumeration constants, 110
  - ISO/ANSI C++, 56–57
  - returning, 263–268
  - variable, 58–60, 705–707
- values, comparing, 117–134**
  - conditional operators, 130–131
  - extended `if` statements, 122–124
  - generally, 117–119
  - `if` statements, 119–120
  - logical operators and expressions, 126–129
  - nested `if` statements, 120–122
  - nested `if-else` statements, 124–126
  - `switch` statements, 131–134
  - unconditional branching, 134
- values types, storing in a list, 684–685**
- variable declarations**
  - defined, 44, 50–51
  - positioning, 89
- variable numbers, of arguments**
  - accepting, 274–275
  - defined, 260–262
- variable values**
  - changing, 707
  - inspecting, 705–707
  - with specific sets of variables, 58–60
- variables**
  - assignment statements, 77–78
  - Boolean type, 55
  - casting, 76–79
  - data types and. *See* data types
  - declaring, 50–51
  - defined, 48–49
  - explicit casts, 78–79
  - exporting from DLLs, 1108
  - floating-point numbers, 55–56
  - initial values for, 51
  - integer type, 52–53
  - keywords, 50
  - modifying, 71–72
  - naming, 49
  - old-style casts and, 79
  - operands, rules for casting, 76–77
  - sharing memory between, 417
  - unions between, 417–419
  - values, 58–60, 705–707
- .vb (Visual Basic Extension), 1095**
- vector containers**
  - accessing elements in, 615–616
  - capacity and size of, 611–615
  - creating, 608–610
  - inserting/deleting elements, 616–618
  - sorting elements, 623–624
  - storing class objects, 618–622
  - storing pointers, 624–626
- vector containers, STL/CLR, 678–680**
- versions, of programs, 20–21**
- View class, 851–852**
- View menu, Toolbox window, 836**
- viewport, coordinates and, 1009**
- views**
  - activation, 1154–1155
  - updating multiple, 935–937
- views, database**
  - creating, 1188–1193
  - customer view class, 1158–1160
  - dialogs for, 1134–1136
  - multiple table views, 1150–1151
  - record view class, adding, 1143–1147
  - record view class, initializing, 1149–1150
  - record view class, overview of, 1132–1134
  - switching, 1151–1155
- views, MFC**
  - linking documents and views in MFC, 781–782
  - overview of, 780–781
- virtual destructors**
  - calling wrong destructor, 545–548
  - how it works, 548–550
  - overview of, 544, 544–545
- virtual functions**
  - abstract classes, 539–542
  - exercises, 597–599
  - incomplete class definitions, 538
  - indirect base classes, 542–544
  - overview of, 530–531
  - pointers to class objects, 535–537
  - polymorphism and, 535
  - pure virtual functions, 538–539
  - references with, 537–538
  - summary, 596–597
  - virtual destructors, 544–550
  - what it is, 532–535
  - working with inherited functions, 531–532
- virtual functions, C++/CLI, 555**
- virtual keyword, 532**
- virtual machine environments, 2**
- virtual mode, DataGridView controls, 1269**

## visibility specifiers

### visibility specifiers

- cancel button and, 1180–1181
- for classes and interfaces, 565

### Visual Basic Extension (.vb), 1095

### Visual C++ 2008, overview, 1–34

- applications, writing, 3–4
- attributes, 6
- CLR and, 2–3
- CLR console, 24–26
- console application files, 19
- console applications, 6–7
- debugging and releasing program versions, 20
- dockable toolbars, 12–13
- documentation, 13
- empty consoles, 21–23
- error handling, 23–24
- executing program versions, 20–21
- IDE for. See IDE (Integrated Development Environment)
- introduction, 1
- learning about, 4–9
- MFC-based applications, 27–30
- .NET framework, 2
- options, setting, 26–27
- projects, 13–14
- solutions, 19
- source code, 17–19
- standards, 5–6
- summary, 33–34
- toolbar options, 11–12
- Win32 console applications, 14–19
- Windows applications, 27–30
- Windows Forms applications, 30–33
- Windows programming concepts and, 7–9

## W

### watch, setting for variables, 705

### Watch windows, debugger, 707

### Web services, as data source, 1266

### WebBrowser control, 1226–1227

### WHERE clause, in SELECT statements, 1117

### while loops, 146–148

### whitespace, 46–47

### Win32 API, 1095

### Win32 console applications, 7, 14–19

### WINDCLASSEX structure, 753–755

### window caption, modifying, 1139–1140

### window class (CFrameWnd), MFC, 773

### WindowProc()

- code for complete function, 767–768

### decoding Windows messages, 764–765

### organization of Windows programs and, 769–770

### processing Windows messages, 763–764

### redrawing client area, 758

### redrawing client area for Windows programs, 765–766

### structure of Windows programs and, 750

### WindowProc(), 766

### windows, in Windows OSs

#### creating program windows, 756–757

#### elements of, 744–746

#### initializing program windows, 757–758

#### structure for defining program windows, 753–755

#### target windows for Windows messages, 1317–1318

#### terminology regarding, 746

### Windows API

#### creating simple, 768–769

#### drawing process, 847–848

#### methods for creating interactive Windows application, 744

#### overview of, 747–748

### Windows applications

#### basic classes, 783

#### creating and executing, 27–30

#### MDI (multiple document interface) application, 801–802

#### MFC tools for creating, 784–785

#### MFC-based, 786–790

#### process of creating, 785–786

#### SDI (single document interface) application, creating, 786–790

### Windows applications, CLR-based

#### event handlers, implementing, 839–840

#### menu items, adding event handlers for, 838–839

#### menu items, setting checks, 840–842

#### menus, adding, 836–838

#### summary, 845

#### toolbars, adding, 842–844

### Windows applications, MFC-based, 786–790

#### adding menus to, 811

#### command messages, processing, 810–811

#### completing the menu, 815

#### creating, 785–786

#### data members, adding to class definitions, 820–821

#### data members, initializing new, 821–823

#### handling messages, 810

#### handling Windows messages, 805–806

#### interactive, 744

#### MDI applications, 801–802

#### menu items, adding to Element menu, 814–815

- menu items, adding to menu bar, 813
- menu items, modifying existing, 815
- menu messages, classes for, 817
- menu messages, coding functions for, 820
- menu messages, creating functions for, 817–820
- menu messages, event handlers for, 816
- menu resources, creating and editing, 812
- message categories, 809–810
- message handler definitions, 807–809
- message handlers, coding update handler, 825–827
- message handlers, exercising update handler, 827
- message handlers, for updating user interface, 824–825
- message maps, 806–807
- running, 802, 823–824
- summary, 845
- toolbar buttons, adding, 827–828
- toolbar buttons, editing properties of, 829–830
- toolbar buttons, exercising, 830
- tools for creating, 784–785
- tooltips, adding, 830

**Windows client area, 765–766. See also client area**  
**Windows Forms Application template, 832**

#### **Windows Forms applications, 1217–1264**

- Button controls, 1224–1226
- buttons, click events, 1238–1239
- buttons, event handlers, 1253–1256
- Choose menu item, click events, 1260–1263
- Choose menu item, dealing with, 1257
- context menus, adding, 1228
- context menus, responding to, 1256–1257
- creating, 30–33
- dialog boxes, adding second, 1245–1248
- dialog boxes, creating, 1235–1236
- dialog boxes, getting data from, 1248–1251
- dialog boxes, using, 1240–1242
- dialog class, developing, 1258–1260
- dialog form, creating, 1257–1258
- dialog objects, creating, 1239–1240
- event handlers, creating, 1228–1229
- event handlers, for buttons, 1231–1235
- event handlers, for Limits menu, 1235, 1252–1253
- event handlers, for menu items, 1229–1230
- event handlers, for Play menu event, 1231–1235
- example, operation of, 1227–1228
- exercise creating, 775–777
- exercises, 1264
- GroupBox controls, using, 1222–1224
- GUI, creating, 1217–1218
- Help>About menu items, 1253

- input controls, disabling, 1251
- input validation, 1242–1244
- ListBox controls, adding list to, 1237–1238
- ListBox objects, state of, 1239
- members, adding to Form1 class, 1230–1231
- methods for creating interactive Windows application, 744
- operations in development of, 831–832
- outputting trace information from, 741
- overview of, 775, 832–834
- properties, 834–835
- Reset menu item, handling, 1245
- start-up, 835
- submenus, 1219–1220
- summary, 1263
- tab controls, adding, 1220–1222
- WebBrowser control, using, 1226–1227

#### **Windows Forms, data source access, 1265–1305**

- accessing/displaying data, 1267
- binding to individual controls, 1299–1302
- BindingNavigator component, 1296–1297
- BindingSource component, 1290–1296
- cell styles, dynamically setting, 1284–1289
- column formats, 1281–1283
- column headers, setting up, 1280–1281
- control, setting up, 1279–1280
- data, setting up, 1278–1279
- DataGridView controls, 1267–1269
- DataGridView controls, customizing, 1275–1277
- DataGridView controls, in bound mode, 1290
- DataGridView controls, in unbound mode, 1269–1275
- overview of, 1265
- rows, customizing alternate, 1283–1284
- summary, 1302–1304
- tables, working with multiple, 1302–1304
- updating database, 1297–1298
- working with data sources, 1266

#### **Windows Forms, library for, 10**

#### **Windows messages**

- message categories, 809–810
- MFC handling, 805–806, 810
- overview of, 747
- processing functions, 763–765
- queues, 758–761
- types of, 1317–1318

#### **Windows programming, 743–778. See also**

- WinMain()
- concepts, 7–9
- data types, 748–749
- DLLs and, 1095

## Windows programming (*continued*)

---

### Windows programming (*continued*)

- ending programs, 766–767
- event-driven programming and, 746
- message processing functions, 763–765
- message queues, 758–761
- messages, 747
- methods for creating interactive Windows application, 744
- MFC for. *See* MFC (Microsoft Foundation Classes)
- notation, 749–750
- organization of Windows programs, 769–770
- overview of, 743
- program window, creating, 756–757
- program window, initializing, 757–758
- program window, specifying, 753–755
- structure of a Windows program, 750–751
- summary, 778
- Window client area, drawing, 765–766
- window elements, 744–746
- WindowProc(), 767–768
- Windows API, 747–748, 768–769
- Windows Forms and, 775–777
- Windows OS and, 746
- Windows Sockets, MFC extension DLLs, 1101**

- WinMain()
  - complete function assembled, 761–763
  - messages, retrieving for program, 758–761
  - organization of Windows programs and, 769–770
  - overview of, 751–753
  - program window, creating, 756–757
  - program window, initializing, 757–758
  - program window, specifying, 753–755
  - structure of Windows programs and, 750
- WM\_COMMAND, **Windows messages, 808**
- WM\_LBUTTONDOWN, **Windows messages, 865**
- WM\_LBUTTONDOWN, **Windows messages, 865, 891–892**
- WM\_MOUSEMOVE, **Windows messages, 865, 885**
- WM\_MOUSEMOVE, **Windows messages, 959–961**
- WM\_PAINT, **Windows messages, 848**
- WM\_RBUTTONDOWN, **Windows messages, 963**
- WndProc(), **handling Windows message, 747**
- Write(), **generating debugging output, 732–733**
- WriteLine(), **generating debugging output, 732–733**
- write-only properties, 390**

## X

- XML serialization, 1078**