Join the discussion @ p2p.wrox.com

Wrox Programmer to Programmer™

wrox
A Wiley Brand

Ivor Horton's

Beginning Visual C++® 2013

Ivor Horton

IVOR HORTON'S BEGINNING VISUAL C++® 2013

INTRODUCTION xxxiii

CHAPTER 1 Programming with Visual C++ 1

CHAPTER 2 Data, Variables, and Calculations. 25

CHAPTER 3 Decisions and Loops 91

CHAPTER 4 Arrays, Strings, and Pointers 129

CHAPTER 5 Introducing Structure into Your Programs. 179

CHAPTER 6 More about Program Structure 221

CHAPTER 7 Defining Your Own Data Types 267

CHAPTER 8 More on Classes. 323

CHAPTER 9 Class Inheritance and Virtual Functions. 431

CHAPTER 10 The Standard Template Library 479

CHAPTER 11 Windows Programming Concepts. 579

CHAPTER 12 Windows Programming with the Microsoft Foundation Classes (MFC) 613

CHAPTER 13 Working with Menus and Toolbars 637

CHAPTER 14 Drawing in a Window. 663

CHAPTER 15 Improving the View 717

CHAPTER 16 Working with Dialogs and Controls. 747

CHAPTER 17 Storing and Printing Documents 793

CHAPTER 18 Programming for Windows 8. 827

INDEX 881

IVOR HORTON'S
BEGINNING

Visual C++® 2013

IVOR HORTON'S
BEGINNING

Visual C++® 2013

Ivor Horton

wrox™
A Wiley Brand

Ivor Horton's Beginning Visual C++® 2013

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2014 by Ivor Horton

Published by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-84571-4
ISBN: 978-1-118-84568-4 (ebk)
ISBN: 978-1-118-84577-6 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2014930412

Trademarks: Wiley, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Visual C++ is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

This book is for my dear wife, Eve.

ABOUT THE AUTHOR

IVOR HORTON graduated as a mathematician and was lured into information technology by promises of great rewards for very little work. In spite of the reality usually being a great deal of work for relatively modest rewards, he has continued to work with computers to the present day. He has been engaged at various times in programming, systems design, consultancy, and the management and implementation of projects of considerable complexity.

Horton has many years of experience in the design and implementation of computer systems applied to engineering design and manufacturing operations in a variety of industries. He has considerable experience in developing occasionally useful applications in a wide variety of programming languages, and in teaching primarily scientists and engineers to do likewise. He has been writing books on programming for several years, and his currently published works include tutorials on C, C++, and Java. At the present time, when he is not writing programming books or providing advice to others, he spends his time fishing, traveling, and enjoying life in general.

ABOUT THE TECHNICAL EDITORS

GIOVANNI DICANIO is a Microsoft Visual C++ MVP, computer programmer, and Pluralsight author. His computer programming experience dates back to the glorious Commodore 64 and Commodore Amiga 500 golden days. He started with C=64 BASIC, then moved to assembly, Pascal, C, C++, Java, and C#. Giovanni wrote computer programming articles on C++, MFC, OpenGL, and other programming subjects in Italian computer magazines. He contributed code to some open-source projects as well, including a mathematical expression parser written in C++ for one of the first versions of QCAD. Giovanni's programming experience includes Windows programming using C++, Win32, COM, and ATL. His favorite programming languages are C and C++.

He has recently started cultivating an interest for mobile platforms and embedded systems.

He can be contacted via e-mail at `giovanni.dicanio@gmail.com`.

MARC GREGOIRE is a software engineer from Belgium. He graduated from the University of Leuven, Belgium, with a degree in "Burgerlijk ingenieur in de computer wetenschappen" (equivalent to Master of Science in engineering in computer science). The year after, he received the cum laude degree of master in artificial intelligence at the same university. After his studies, Marc started working for a software consultancy company called Ordina Belgium. As a consultant, he worked for Siemens and Nokia Siemens Networks on critical 2G and 3G software running on Solaris for telecom operators. This required working in international teams stretching from South America and the United States to EMEA and Asia. Now, Marc is working for Nikon Metrology on 3D laser scanning software.

His main expertise is C/C++, and specifically Microsoft VC++ and the MFC framework. He has experience in developing C++ programs running 24x7 on Windows and Linux platforms; for example, KNX/EIB home automation software. Next to C/C++, Marc also likes C# and uses PHP for creating web pages.

Since April 2007, he received the yearly Microsoft MVP (Most Valuable Professional) award for his Visual C++ expertise.

Marc is the founder of the Belgian C++ Users Group (`www.becpp.org`), author of *Professional C++*, Wrox, 2011 (ISBN 978-047-0-93244-9) and a member on the CodeGuru forum (as Marc G). He maintains a blog on `www.nuonsoft.com/blog/`.

CREDITS

Executive Editor
Robert Elliott

Project Editors
Sydney Jones Argenta
Edward Connor

Technical Editors
Giovani Dicanio
Marc Gregoire

Production Editor
Christine Mugnolo

Copy Editor
Charlotte Kughen

Manager of Content Development and Assembly
Mary Beth Wakefield

Director of Community Marketing
David Mayhew

Marketing Manager
Ashley Zurcher

Business Manager
Amy Knies

Vice President and Executive Group Publisher
Richard Swadley

Associate Publisher
Jim Minatel

Project Coordinator, Cover
Todd Klemme

Proofreader
Sarah Kaikini, Word One New York

Indexer
Johnna VanHoose Dinse

Cover Designer
Wiley

Cover Image
©iStockphoto.com/xyno

ACKNOWLEDGMENTS

THE AUTHOR is only one member of the large team of people necessary to get a book into print. I'd like to thank the John Wiley & Sons and Wrox Press editorial and production teams for their help and support throughout.

I would particularly like to thank my technical editors, Marc Gregoire and Giovanni Dicanio, for doing such a fantastic job of reviewing the text and checking out all the code fragments and examples. Their many constructive comments and suggestions have undoubtedly made the book a much better tutorial.

CONTENTS

INTRODUCTION *xxxiii*

CHAPTER 1: PROGRAMMING WITH VISUAL C++ 1

Learning with Visual C++ 1
Writing C++ Applications 2
Learning Desktop Applications Programming 3
Learning C++ 3
C++ Concepts 3
Functions 4
Data and Variables 4
Classes and Objects 4
Templates 4
Program Files 4
Console Applications 5
Windows Programming Concepts 5
The Integrated Development Environment 6
The Editor 7
The Compiler 7
The Linker 7
The Libraries 7
The Standard C++ Library 8
Microsoft Libraries 8
Using the IDE 8
Toolbar Options 9
Dockable Toolbars 10
Documentation 10
Projects and Solutions 10
Defining a Project 11
Debug and Release Versions of Your Program 15
Executing the Program 16
Dealing with Errors 18
Setting Options in Visual C++ 19
Creating and Executing Windows Applications 20
Creating an MFC Application 20
Building and Executing the MFC Application 22
Summary 23

CHAPTER 2: DATA, VARIABLES, AND CALCULATIONS 25

The Structure of a C++ Program 26
Program Comments 31
The #include Directive — Header Files 31
Namespaces and the Using Declaration 32
The main() Function 33
Program Statements 33
Whitespace 35
Statement Blocks 36
Automatically Generated Console Programs 36
Precompiled Header Files 37
Main Function Names 37
Defining Variables 38
Naming Variables 38
Keywords 39
Declaring Variables 39
Initial Values for Variables 40
Fundamental Data Types 40
Integer Variables 41
Character Data Types 42
Integer Type Modifiers 43
The Boolean Type 43
Floating-point Types 44
Fundamental Types in C++ 45
Literals 45
Defining Type Aliases 46
Basic Input/Output Operations 47
Input from the Keyboard 47
Output to the Command Line 48
Formatting the Output 49
Escape Sequences 50
Calculating in C++ 52
The Assignment Statement 52
Arithmetic Operations 52
The const Modifier 55
Constant Expressions 55
Program Input 56
Calculating the Result 56
Displaying the Result 57
Calculating a Remainder 58
Modifying a Variable 58

The Increment and Decrement Operators 59
The Sequence of Calculation 61
Operator Precedence 61
Type Conversion and Casting 63
Type Conversion in Assignments 64
Explicit Type Conversion 64
Old-style Casts 65
The auto Keyword 65
Discovering Types 66
The Bitwise Operators 67
The Bitwise AND 67
The Bitwise OR 68
The Bitwise Exclusive OR 69
The Bitwise NOT 70
The Bitwise Shift Operators 70
Introducing Lvalues and Rvalues 72
Understanding Storage Duration and Scope 72
Automatic Variables 73
Positioning Variable Declarations 75
Global Variables 75
Static Variables 78
Variables with Specific Sets of Values 79
Old Enumerations 79
Type-safe Enumerations 81
Namespaces 83
Declaring a Namespace 84
Multiple Namespaces 86
Summary 86

CHAPTER 3: DECISIONS AND LOOPS 91

Comparing Values 91
The if Statement 93
Nested if Statements 94
The Extended if Statement 96
Nested if-else Statements 98
Logical Operators and Expressions 100
Logical AND 101
Logical OR 101
Logical NOT 102
The Conditional Operator 103
The switch Statement 105

Unconditional Branching 109
Repeating a Block of Statements 109
What Is a Loop? 110
Variations on the for Loop 112
Using the continue Statement 116
Floating-Point Loop Counters 120
The while Loop 120
The do-while Loop 122
The Range-Based for Loop 123
Nested Loops 124
Summary 127

CHAPTER 4: ARRAYS, STRINGS, AND POINTERS 129

Handling Multiple Data Values of the Same Type 129
Arrays 130
Declaring Arrays 131
Initializing Arrays 134
Using the Range-based for Loop 135
Multidimensional Arrays 136
Initializing Multidimensional Arrays 137
Working with C-Style Strings 140
String Input 141
String Literals 143
Using the Range-based for Loop with Strings 144
Indirect Data Access 146
What Is a Pointer? 146
Declaring Pointers 146
The Address-of Operator 147
Using Pointers 147
The Indirection Operator 147
Why Use Pointers? 148
Initializing Pointers 148
Pointers to char 150
The sizeof Operator 154
Constant Pointers and Pointers to Constants 155
Pointers and Arrays 157
Pointer Arithmetic 157
Using Pointers with Multidimensional Arrays 162
Pointer Notation with Multidimensional Arrays 162
Dynamic Memory Allocation 163
The Free Store, Alias the Heap 163

The new and delete Operators 164
Allocating Memory Dynamically for Arrays 165
Dynamic Allocation of Multidimensional Arrays 167
Using References 168
What Is a Reference? 168
Declaring and Initializing Lvalue References 169
Using References in a Range-based for Loop 169
Creating Rvalue References 170
Library Functions for Strings 171
Finding the Length of a Null-terminated String 171
Joining Null-terminated Strings 172
Copying Null-terminated Strings 173
Comparing Null-terminated Strings 173
Searching Null-terminated Strings 174
Summary 176

CHAPTER 5: INTRODUCING STRUCTURE INTO YOUR PROGRAMS 179

Understanding Functions 179
Why Do You Need Functions? 181
Structure of a Function 181
The Function Header 181
The Function Body 183
The return Statement 183
Alternative Function Syntax 184
Using a Function 184
Function Prototypes 184
Passing Arguments to a Function 188
The Pass-by-Value Mechanism 189
Pointers as Arguments to a Function 190
Passing Arrays to a Function 192
Passing Multidimensional Arrays to a Function 194
References as Arguments to a Function 196
Use of the const Modifier 198
Rvalue Reference Parameters 200
Arguments to main() 202
Accepting a Variable Number of Function Arguments 204
Returning Values from a Function 206
Returning a Pointer 206
A Cast-Iron Rule for Returning Addresses 208
Returning a Reference 209
A Cast-Iron Rule: Returning References 212

Static Variables in a Function 212
Recursive Function Calls 214
Using Recursion 217
Summary 217

CHAPTER 6: MORE ABOUT PROGRAM STRUCTURE 221

Pointers to Functions 221
Declaring Pointers to Functions 222
A Pointer to a Function as an Argument 225
Arrays of Pointers to Functions 227
Initializing Function Parameters 228
Exceptions 229
Throwing Exceptions 231
Catching Exceptions 232
Rethrowing Exceptions 234
Exception Handling in the MFC 234
Handling Memory Allocation Errors 235
Function Overloading 237
What Is Function Overloading? 237
Reference Types and Overload Selection 240
When to Overload Functions 241
Function Templates 241
Using a Function Template 241
Using the decltype Operator 244
An Example Using Functions 246
Implementing a Calculator 247
Analyzing the Problem 247
Eliminating Blanks from a String 250
How the Function Functions 250
Evaluating an Expression 250
How the Function Functions 252
Getting the Value of a Term 253
How the Function Functions 254
Analyzing a Number 254
How the Function Functions 256
Putting the Program Together 258
How the Function Functions 259
Extending the Program 259
How the Function Functions 261
Extracting a Substring 261

How the Function Functions 263
Running the Modified Program 263
Summary 264

CHAPTER 7: DEFINING YOUR OWN DATA TYPES 267

The struct in C++ 267
What Is a struct? 268
Defining a struct 268
Initializing a struct 269
Accessing the Members of a struct 269
IntelliSense Assistance with Structures 272
The RECT Structure 274
Using Pointers with a struct 274
Accessing Structure Members through a Pointer 275
The Indirect Member Selection Operator 276
Types, Objects, Classes, and Instances 276
First Class 277
Operations on Classes 278
Terminology 278
Understanding Classes 279
Defining a Class 279
Access Control in a Class 280
Declaring Objects of a Class 280
Accessing the Data Members of a Class 280
Memberwise Initialization of an Object 283
Initializing Class Members 283
Member Functions of a Class 283
Defining a Member Function Outside a Class 286
Inline Functions 286
Class Constructors 287
What Is a Constructor? 287
The Default Constructor 289
Default Parameter Values 292
Using a Constructor Initialization List 294
Making a Constructor Explicit 295
Delegating Constructors 296
Private Members of a Class 297
Accessing private Class Members 299
The friend Functions of a Class 300
Placing friend Function Definitions Inside the Class 302

The Default Copy Constructor 303
The Pointer this 304
Const Objects 306
const Member Functions of a Class 307
Member Function Definitions Outside the Class 308
Arrays of Objects 309
Static Members of a Class 310
Static Data Members 311
Static Function Members of a Class 313
Pointers and References to Objects 314
Pointers to Objects 314
References to Class Objects 317
Implementing a Copy Constructor 317
Summary 318

CHAPTER 8: MORE ON CLASSES 323

Class Destructors 324
What Is a Destructor? 324
The Default Destructor 324
Destructors and Dynamic Memory Allocation 326
Implementing a Copy Constructor 329
Operator Overloading 331
Implementing an Overloaded Operator 332
Implementing Full Support for Comparison Operators 335
Overloading the Assignment Operator 340
Fixing the Problem 340
Overloading the Addition Operator 344
Overloading the Increment and Decrement Operators 348
Overloading the Function Call Operator 350
The Object Copying Problem 351
Avoiding Unnecessary Copy Operations 351
Applying Rvalue Reference Parameters 354
Named Objects Are Lvalues 356
Default Class Members 361
Class Templates 362
Defining a Class Template 363
Template Member Functions 365
Creating Objects from a Class Template 366
Class Templates with Multiple Parameters 370
Templates for Function Objects 372
Perfect Forwarding 373

Default Arguments for Template Parameters 376
Default Function Template Arguments 376
Default Class Template Arguments 378
Aliases for Class Templates 381
Template Specialization 382
Using Classes 385
The Idea of a Class Interface 385
Defining the Problem 386
Implementing the CBox Class 386
Comparing CBox Objects 387
Combining CBox Objects 388
Analyzing CBox Objects 390
Organizing Your Program Code 403
Naming Program Files 404
Library Classes for Strings 405
Creating String Objects 405
Concatenating Strings 407
Accessing and Modifying Strings 410
Comparing Strings 414
Searching Strings 418
Summary 427
CHAPTER 9: CLASS INHERITANCE AND VIRTUAL FUNCTIONS 431
Object-Oriented Programming Basics 432
Inheritance in Classes 433
What Is a Base Class? 433
Deriving Classes from a Base Class 434
Access Control Under Inheritance 437
Constructor Operation in a Derived Class 440
Declaring Protected Class Members 443
The Access Level of Inherited Class Members 446
The Copy Constructor in a Derived Class 447
Preventing Class Derivation 450
Class Members as Friends 451
Friend Classes 453
Limitations on Class Friendship 453
Virtual Functions 453
What Is a Virtual Function? 456
Ensuring Correct Virtual Function Operation 457
Preventing Function Overriding 458
Using Pointers to Class Objects 458

Using References with Virtual Functions 460
Pure Virtual Functions 461
Abstract Classes 462
Indirect Base Classes 464
Virtual Destructors 467
Casting Between Class Types 470
Defining Conversion Operators 470
Explicit Conversion Operators 471
Nested Classes 471
Summary 475

CHAPTER 10: THE STANDARD TEMPLATE LIBRARY 479

What Is the Standard Template Library? 479
Containers 480
Allocators 482
Comparators 482
Container Adapters 483
Iterators 483
Iterator Categories 483
SCARY Iterators 485
Functions Returning Iterators 485
Smart Pointers 485
Using unique_ptr Objects 486
Using shared_ptr Objects 487
Accessing the Raw Pointer in a Smart Pointer 488
Casting Smart Pointers 488
Algorithms 488
Function Objects in the STL 489
Function Adapters 490
The Range of STL Containers 490
Sequence Containers 490
Creating Vector Containers 492
The Capacity and Size of a Vector Container 495
Accessing the Elements in a Vector 499
Inserting and Deleting Elements in a Vector 500
Insert Operations 500
Emplace Operations 501
Erase Operations 502
Swap and Assign Operations 502
Storing Class Objects in a Vector 503
Sorting Vector Elements 508
Storing Pointers in a Vector 509

Array Containers 511
Double-ended Queue Containers 515
Using List Containers 518
Adding Elements to a List 518
Accessing Elements in a List 520
Sorting List Elements 520
Other Operations on Lists 522
Using forward_list Containers 528
Using Other Sequence Containers 529
Queue Containers 530
Priority Queue Containers 532
Stack Containers 537
The tuple Class Template 539
Associative Containers 543
Using Map Containers 543
Storing Objects 544
Accessing Objects 546
Other Map Operations 546
Using a Multimap Container 555
More on Iterators 556
Using Input Stream Iterators 556
Using Inserter Iterators 559
Using Output Stream Iterators 560
More on Function Objects 562
More on Algorithms 564
Type Traits and Static Assertions 566
Lambda Expressions 567
The Capture Clause 568
Capturing Specific Variables 569
Templates and Lambda Expressions 570
Naming a Lambda Expression 574
Summary 577

CHAPTER 11: WINDOWS PROGRAMMING CONCEPTS 579

Windows Programming Basics 580
Elements of a Window 580
Windows Programs and the Operating System 582
Event-Driven Programs 583
Windows Messages 583
The Windows API 583
Windows Data Types 584
Notation in Windows Programs 585

The Structure of a Windows Program 586
The WinMain( ) Function 587
Specifying a Program Window 589
Creating a Program Window 591
Initializing the Program Window 593
Dealing with Windows Messages 594
A Complete WinMain( ) Function 598
How It Works 599
Processing Windows Messages 599
The WindowProc( ) Function 600
Decoding a Windows Message 600
Ending the Program 603
A Complete WindowProc( ) Function 603
How It Works 604
The Microsoft Foundation Classes 605
MFC Notation 605
How an MFC Program Is Structured 606
Summary 610

CHAPTER 12: WINDOWS PROGRAMMING WITH THE MICROSOFT FOUNDATION CLASSES (MFC) 613

The MFC Document/View Concept 614
What Is a Document? 614
Document Interfaces 614
What Is a View? 614
Linking a Document and Its Views 615
Document Templates 616
Document Template Classes 616
Your Application and MFC 617
Creating MFC Applications 618
Creating an SDI Application 619
MFC Application Wizard Output 623
Viewing Project Files 625
Viewing Classes 625
The Class Definitions 626
Creating an Executable Module 631
Running the Program 631
How the Program Works 632
Creating an MDI Application 633
Running the Program 634
Summary 635

CHAPTER 13: WORKING WITH MENUS AND TOOLBARS 637

Communicating with Windows 637
Understanding Message Maps 638
Message Handler Definitions 639
Message Categories 640
Handling Messages in Your Program 641
How Command Messages Are Processed 642
Extending the Sketcher Program 642
Elements of a Menu 643
Creating and Editing Menu Resources 643
Adding a Menu Item to the Menu Bar 644
Adding Items to the Element Menu 645
Modifying Existing Menu Items 645
Completing the Menu 646
Adding Menu Message Handlers 646
Choosing a Class to Handle Menu Messages 648
Creating Menu Message Handlers 648
Implementing Menu Message Handlers 650
Adding Members to Store Color and Element Mode 650
Defining Element and Color Types 652
Initializing the Color and Element Type Members 653
Implementing Menu Command Message Handlers 653
Running the Extended Example 654
Adding Menu Update Message Handlers 654
Coding a Command Update Handler 655
Exercising the Update Handlers 656
Adding Toolbar Buttons 656
Editing Toolbar Button Properties 658
Exercising the Toolbar Buttons 659
Adding Tooltips 659
Summary 660

CHAPTER 14: DRAWING IN A WINDOW 663

Basics of Drawing in a Window 663
The Window Client Area 664
The Windows Graphical Device Interface 664
Working with a Device Context 665
Mapping Modes 665
The MFC Drawing Mechanism 667
The View Class in Your Application 667

The OnDraw() Member Function 668
The CDC Class 669
Displaying Graphics 669
Drawing in Color 673
Drawing Graphics in Practice 678
Programming for the Mouse 680
Messages from the Mouse 681
WM_LBUTTONDOWN 682
WM_MOUSEMOVE 682
WM_LBUTTONUP 682
Mouse Message Handlers 682
Drawing Using the Mouse 684
Getting the Client Area Redrawn 686
Defining Element Classes 687
The CElement Class 691
The CLine Class 692
The CRectangle Class 695
The CCircle Class 697
The CCurve Class 700
Completing the Mouse Message Handlers 702
Drawing a Sketch 709
Running the Example 710
Capturing Mouse Messages 711
Summary 713

CHAPTER 15: IMPROVING THE VIEW 717

Sketcher Limitations 717
Improving the View 718
Updating Multiple Views 718
Scrolling Views 720
Logical Coordinates and Client Coordinates 722
Dealing with Client Coordinates 723
Using MM_LOENGLISH Mapping Mode 725
Deleting and Moving Elements 726
Implementing a Context Menu 726
Associating a Menu with a Class 728
Checking Context Menu Items 729
Identifying an Element Under the Cursor 730
Exercising the Context Menus 732
Highlighting Elements 732
Drawing Highlighted Elements 735