

**Ivor Horton's
Beginning
Visual C++[®] 2008**

Ivor Horton



WILEY

Wiley Publishing, Inc.

Ivor Horton's Beginning Visual C++[®] 2008

Introduction	xxv
Chapter 1: Programming with Visual C++ 2008	1
Chapter 2: Data, Variables, and Calculations.	35
Chapter 3: Decisions and Loops	117
Chapter 4: Arrays, Strings, and Pointers	161
Chapter 5: Introducing Structure into Your Programs	239
Chapter 6: More about Program Structure	279
Chapter 7: Defining Your Own Data Types	331
Chapter 8: More on Classes.	409
Chapter 9: Class Inheritance and Virtual Functions.	507
Chapter 10: The Standard Template Library	601
Chapter 11: Debugging Techniques	695
Chapter 12: Windows Programming Concepts	743
Chapter 13: Windows Programming with the Microsoft Foundation Classes	779
Chapter 14: Working with Menus and Toolbars	805
Chapter 15: Drawing in a Window.	847
Chapter 16: Creating the Document and Improving the View.	909
Chapter 17: Working with Dialogs and Controls	985
Chapter 18: Storing and Printing Documents	1047
Chapter 19: Writing Your Own DLLs	1093
Chapter 20: Connecting to Data Sources	1113
Chapter 21: Updating Data Sources	1167
Chapter 22: More on Windows Forms Applications	1217
Chapter 23: Accessing Data Sources in a Windows Forms Application	1265
Appendix A: C++ Keywords	1307
Appendix B: ASCII Codes.	1309
Appendix C: Windows Message Types	1317
Index	1319

**Ivor Horton's
Beginning
Visual C++[®] 2008**

Ivor Horton



Wiley Publishing, Inc.

Ivor Horton's Beginning Visual C++® 2008

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2008 by Ivor Horton
Published by Wiley Publishing, Inc., Indianapolis, Indiana
Published simultaneously in Canada
ISBN: 978-0-470-22590-5

Manufactured in the United States of America
10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data is available from the publisher.

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 Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, 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 Website 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 Website may provide or recommendations it may make. Further, readers should be aware that Internet Websites 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 (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Wrox 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 in the United States and/or other countries. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

This book is dedicated to Isabella Williams, who joined the growing band of beautiful girls in my life in January 2007.

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 being usually 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 of the 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 to manufacturing operations in a variety of industries. He has considerable experience developing occasionally useful applications in a wide variety of programming languages, and teaching primarily scientists and engineers to do likewise. He has been writing books on programming for more than 10 years now, 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 trying to speak better French.

Acknowledgments

I'd like to acknowledge the efforts and support of the John Wiley & Sons and Wrox Press editorial and production team in the production of this book, especially my development editor, Ed Connor, who has been there from way back at the beginning and has stayed through to the end. I'd also like to thank my technical editor, John Mueller, once again for doing such an outstanding job of reviewing the text and checking out all the examples in the book; his many constructive comments undoubtedly helped make the book a much better tutorial.

As always, the love and support of my wife, Eve, have been fundamental to making it possible for me to write this book. She has pandered to my every need throughout and has remained patient and cheerful in spite of the hours I spend locked away in my office and my intermittent grumblings about the workload I imposed upon myself.

Credits

Executive Editor

Robert Elliott

Development Editor

Ed Connor

Technical Editor

John Mueller

Copy Editor

Mildred Sanchez

Editorial Manager

Mary Beth Wakefield

Production Manager

Tim Tate

Vice President and Executive Group Publisher

Richard Swadley

Vice President and Executive Publisher

Joseph B. Wikert

Project Coordinator, Cover

Lynsey Stanford

Composer

Laurie Stewart, Happenstance Type-O-Rama

Proofreaders

Jeremy Bagai

Sheilah Ledwidge

Carrie Hunter

Lee Ewert

Will DeRooy

Indexer

Jack Lewis

Contents

Acknowledgments	v
Introduction	xxv
Chapter 1: Programming with Visual C++ 2008	1
The .NET Framework	2
The Common Language Runtime (CLR)	2
Writing C++ Applications	3
Learning Windows Programming	4
Learning C++	5
The C++ Standards	5
Attributes	6
Console Applications	6
Windows Programming Concepts	7
What Is the Integrated Development Environment?	9
Components of the System	9
Using the IDE	10
Toolbar Options	11
Dockable Toolbars	12
Documentation	13
Projects and Solutions	13
Setting Options in Visual C++ 2008	26
Creating and Executing Windows Applications	27
Creating a Windows Forms Application	30
Summary	33
Chapter 2: Data, Variables, and Calculations	35
The Structure of a C++ Program	36
The main() Function	44
Program Statements	44
Whitespace	46
Statement Blocks	47
Automatically Generated Console Programs	47
Defining Variables	48
Naming Variables	49

Contents

Declaring Variables	50
Initial Values for Variables	51
Fundamental Data Types	51
Integer Variables	52
Character Data Types	53
Integer Type Modifiers	54
The Boolean Type	55
Floating-Point Types	55
Literals	57
Defining Synonyms for Data Types	58
Variables with Specific Sets of Values	58
Basic Input/Output Operations	60
Input from the Keyboard	60
Output to the Command Line	60
Formatting the Output	61
Escape Sequences	63
Calculating in C++	65
The Assignment Statement	65
Arithmetic Operations	65
Calculating a Remainder	70
Modifying a Variable	71
The Increment and Decrement Operators	72
The Sequence of Calculation	74
Variable Types and Casting	76
Rules for Casting Operands	76
Casts in Assignment Statements	77
Explicit Casts	78
Old-Style Casts	79
The Bitwise Operators	79
The Bitwise AND	80
The Bitwise OR	82
The Bitwise Exclusive OR	83
The Bitwise NOT	84
The Bitwise Shift Operators	84
Understanding Storage Duration and Scope	86
Automatic Variables	86
Positioning Variable Declarations	89
Global Variables	89
Static Variables	93
Namespaces	93
Declaring a Namespace	94
Multiple Namespaces	96

C++/CLI Programming	97
C++/CLI Specific: Fundamental Data Types	98
C++/CLI Output to the Command Line	102
C++/CLI Specific — Formatting the Output	103
C++/CLI Input from the Keyboard	106
Using safe_cast	107
C++/CLI Enumerations	107
Summary	112
Exercises	114
Chapter 3: Decisions and Loops	117
Comparing Values	117
The if Statement	119
Nested if Statements	120
Nested if-else Statements	124
Logical Operators and Expressions	126
The Conditional Operator	130
The switch Statement	131
Unconditional Branching	134
Repeating a Block of Statements	135
What Is a Loop?	135
Variations on the for Loop	138
The while Loop	146
The do-while Loop	148
Nested Loops	149
C++/CLI Programming	153
The for each Loop	156
Summary	159
Exercises	159
Chapter 4: Arrays, Strings, and Pointers	161
Handling Multiple Data Values of the Same Type	162
Arrays	162
Declaring Arrays	163
Initializing Arrays	166
Character Arrays and String Handling	168
Multidimensional Arrays	171
Indirect Data Access	174
What Is a Pointer?	174
Declaring Pointers	175

Contents

Using Pointers	176
Initializing Pointers	178
The sizeof Operator	183
Constant Pointers and Pointers to Constants	185
Pointers and Arrays	187
Dynamic Memory Allocation	194
The Free Store, Alias the Heap	194
The new and delete Operators	195
Allocating Memory Dynamically for Arrays	195
Dynamic Allocation of Multidimensional Arrays	198
Using References	199
What Is a Reference?	199
Declaring and Initializing References	199
Native C++ Library Functions for Strings	200
Finding the Length of a Null-Terminated String	200
Joining Null-Terminated Strings	201
Copying Null-Terminated Strings	203
Comparing Null-Terminated Strings	204
Searching Null-Terminated Strings	204
C++/CLI Programming	206
Tracking Handles	207
CLR Arrays	209
Strings	224
Tracking References	233
Interior Pointers	233
Summary	236
Exercises	238
Chapter 5: Introducing Structure into Your Programs	239
Understanding Functions	239
Why Do You Need Functions?	241
Structure of a Function	241
Using a Function	243
Passing Arguments to a Function	247
The Pass-by-value Mechanism	247
Pointers as Arguments to a Function	249
Passing Arrays to a Function	251
References as Arguments to a Function	255
Use of the const Modifier	257

Arguments to main()	258
Accepting a Variable Number of Function Arguments	260
Returning Values from a Function	263
Returning a Pointer	263
Returning a Reference	266
Static Variables in a Function	269
Recursive Function Calls	271
Using Recursion	273
C++/CLI Programming	274
Functions Accepting a Variable Number of Arguments	274
Arguments to main()	275
Summary	277
Exercises	277
Chapter 6: More about Program Structure	279
<hr/>	
Pointers to Functions	279
Declaring Pointers to Functions	280
A Pointer to a Function as an Argument	283
Arrays of Pointers to Functions	285
Initializing Function Parameters	285
Exceptions	287
Throwing Exceptions	289
Catching Exceptions	290
Exception Handling in the MFC	291
Handling Memory Allocation Errors	292
Function Overloading	293
What Is Function Overloading?	294
When to Overload Functions	296
Function Templates	296
Using a Function Template	297
An Example Using Functions	299
Implementing a Calculator	299
Eliminating Blanks from a String	302
Evaluating an Expression	303
Getting the Value of a Term	305
Analyzing a Number	306
Putting the Program Together	309
Extending the Program	311
Extracting a Substring	312
Running the Modified Program	314

Contents

C++/CLI Programming	315
Understanding Generic Functions	316
A Calculator Program for the CLR	322
Summary	328
Exercises	329
Chapter 7: Defining Your Own Data Types	331
The struct in C++	332
What Is a struct?	332
Defining a struct	332
Initializing a struct	333
Accessing the Members of a struct	333
IntelliSense Assistance with Structures	337
The struct RECT	338
Using Pointers with a struct	338
Data Types, Objects, Classes, and Instances	340
First Class	342
Operations on Classes	342
Terminology	343
Understanding Classes	343
Defining a Class	344
Declaring Objects of a Class	344
Accessing the Data Members of a Class	345
Member Functions of a Class	347
Positioning a Member Function Definition	349
Inline Functions	350
Class Constructors	351
What Is a Constructor?	351
The Default Constructor	353
Assigning Default Parameter Values in a Class	355
Using an Initialization List in a Constructor	358
Private Members of a Class	358
Accessing private Class Members	361
The friend Functions of a Class	362
The Default Copy Constructor	364
The Pointer this	366
const Objects of a Class	368
const Member Functions of a Class	369
Member Function Definitions Outside the Class	370
Arrays of Objects of a Class	371

Static Members of a Class	373
Static Data Members of a Class	373
Static Function Members of a Class	376
Pointers and References to Class Objects	376
Pointers to Class Objects	376
References to Class Objects	379
C++/CLI Programming	381
Defining Value Class Types	381
Defining Reference Class Types	386
Defining a Copy Constructor for a Reference Class Type	389
Class Properties	390
initonly Fields	403
Static Constructors	405
Summary	405
Exercises	406
Chapter 8: More on Classes	409
Class Destructors	409
What Is a Destructor?	409
The Default Destructor	410
Destructors and Dynamic Memory Allocation	412
Implementing a Copy Constructor	415
Sharing Memory Between Variables	417
Defining Unions	417
Anonymous Unions	419
Unions in Classes and Structures	419
Operator Overloading	420
Implementing an Overloaded Operator	420
Implementing Full Support for a Comparison Operator	424
Overloading the Assignment Operator	427
Overloading the Addition Operator	432
Overloading the Increment and Decrement Operators	436
Class Templates	438
Defining a Class Template	438
Creating Objects from a Class Template	441
Class Templates with Multiple Parameters	444
Using Classes	446
The Idea of a Class Interface	446
Defining the Problem	446
Implementing the CBox Class	447

Contents

Organizing Your Program Code	468
Naming Program Files	470
Native C++ Library Classes for Strings	471
Creating String Objects	471
Concatenating Strings	473
Accessing and Modifying Strings	476
Comparing Strings	480
Searching Strings	484
C++/CLI Programming	494
Overloading Operators in Value Classes	494
Overloading the Increment and Decrement Operators	499
Overloading Operators in Reference Classes	500
Implementing the Assignment Operator for Reference Types	503
Summary	503
Exercises	504
Chapter 9: Class Inheritance and Virtual Functions	507
Basic Ideas of OOP	507
Inheritance in Classes	509
What Is a Base Class?	509
Deriving Classes from a Base Class	510
Access Control Under Inheritance	513
Constructor Operation in a Derived Class	516
Declaring Class Members to Be Protected	519
The Access Level of Inherited Class Members	522
The Copy Constructor in a Derived Class	524
Class Members as Friends	528
Friend Classes	530
Limitations on Class Friendship	530
Virtual Functions	530
What Is a Virtual Function?	532
Using Pointers to Class Objects	535
Using References with Virtual Functions	537
Pure Virtual Functions	538
Abstract Classes	539
Indirect Base Classes	542
Virtual Destructors	544
Casting Between Class Types	550
Nested Classes	550
C++/CLI Programming	553
Boxing and Unboxing	554

Inheritance in C++/CLI Classes	555
Interface Classes	561
Defining Interface Classes	561
Classes and Assemblies	565
Functions Specified as new	570
Delegates and Events	571
Destructors and Finalizers in Reference Classes	583
Generic Classes	585
Summary	596
Exercises	597

Chapter 10: The Standard Template Library **601**

What Is the Standard Template Library?	601
Containers	602
Container Adapters	603
Iterators	604
Algorithms	605
Function Objects	605
Function Adapters	606
The Range of STL Containers	606
Sequence Containers	607
Creating Vector Containers	608
The Capacity and Size of a Vector Container	611
Accessing the Elements in a Vector	615
Inserting and Deleting Elements in a Vector	616
Storing Class Objects in a Vector	618
Sorting Vector Elements	623
Storing Pointers in a Vector	624
Double-Ended Queue Containers	626
Using List Containers	630
Using Other Sequence Containers	640
Associative Containers	651
Using Map Containers	652
Using a Multimap Container	664
More on Iterators	665
Using Input Stream Iterators	665
Using Inserter Iterators	669
Using Output Stream Iterators	670
More on Function Objects	672
More on Algorithms	674
fill()	674

Contents

replace()	674
find()	675
transform()	675
The STL for C++/CLI Programs	676
STL/CLR Containers	677
Using Sequence Containers	677
Using Associative Containers	685
Summary	692
Exercises	693
Chapter 11: Debugging Techniques	695
Understanding Debugging	695
Program Bugs	696
Common Bugs	697
Basic Debugging Operations	698
Setting Breakpoints	700
Setting Tracepoints	702
Starting Debugging	703
Changing the Value of a Variable	707
Adding Debugging Code	708
Using Assertions	708
Adding Your Own Debugging Code	709
Debugging a Program	715
The Call Stack	715
Step Over to the Error	716
Testing the Extended Class	720
Finding the Next Bug	722
Debugging Dynamic Memory	723
Functions Checking the Free Store	723
Controlling Free Store Debug Operations	725
Free Store Debugging Output	726
Debugging C++/CLI Programs	731
Using the Debug and Trace Classes	732
Getting Trace Output in Windows Forms Applications	741
Summary	741
Chapter 12: Windows Programming Concepts	743
Windows Programming Basics	744
Elements of a Window	744
Windows Programs and the Operating System	746

Event-Driven Programs	746
Windows Messages	747
The Windows API	747
Windows Data Types	748
Notation in Windows Programs	749
The Structure of a Windows Program	750
The WinMain() Function	751
Message Processing Functions	763
A Simple Windows Program	768
Windows Program Organization	769
The Microsoft Foundation Classes	770
MFC Notation	771
How an MFC Program Is Structured	771
Using Windows Forms	775
Summary	778
Chapter 13: Windows Programming with the Microsoft Foundation Classes	779
The Document/View Concept in MFC	780
What Is a Document?	780
Document Interfaces	780
What Is a View?	780
Linking a Document and Its Views	781
Your Application and MFC	783
Creating MFC Applications	784
Creating an SDI Application	786
The Output from the MFC Application Wizard	790
Creating an MDI Application	801
Summary	803
Exercises	803
Chapter 14: Working with Menus and Toolbars	805
Communicating with Windows	805
Understanding Message Maps	806
Message Categories	809
Handling Messages in Your Program	810
Extending the Sketcher Program	811
Elements of a Menu	812
Creating and Editing Menu Resources	812
Adding Handlers for Menu Messages	816
Choosing a Class to Handle Menu Messages	817
Creating Menu Message Functions	817

Contents

Coding Menu Message Functions	820
Adding Message Handlers to Update the User Interface	824
Adding Toolbar Buttons	827
Editing Toolbar Button Properties	829
Exercising the Toolbar Buttons	830
Adding Tooltips	831
Menu and Toolbars in a C++/CLI Program	831
Understanding Windows Forms	831
Understanding Windows Forms Applications	832
Adding a Menu to CLR Sketcher	836
Adding Event Handlers for Menu Items	838
Implementing Event Handlers	839
Setting Menu Item Checks	840
Adding a Toolbar	842
Summary	845
Exercises	846
Chapter 15: Drawing in a Window	847
<hr/>	
Basics of Drawing in a Window	847
The Window Client Area	848
The Windows Graphical Device Interface	848
The Drawing Mechanism in Visual C++	851
The View Class in Your Application	851
The CDC Class	852
Drawing Graphics in Practice	861
Programming the Mouse	863
Messages from the Mouse	864
Mouse Message Handlers	865
Drawing Using the Mouse	867
Exercising Sketcher	892
Running the Example	892
Capturing Mouse Messages	893
Drawing with the CLR	895
Drawing on a Form	895
Adding Mouse Event Handlers	895
Defining C++/CLI Element Classes	897
Implementing the MouseMove Event Handler	904
Implementing the MouseUp Event Handler	905
Implementing the Paint Event Handler for the Form	905
Summary	906
Exercises	907

Chapter 16: Creating the Document and Improving the View	909
The MFC Collection Classes	909
Types of Collection	910
The Type-Safe Collection Classes	911
Collections of Objects	911
The Typed Pointer Collections	920
Using the CList Template Class	923
Drawing a Curve	924
Defining the CCurve Class	925
Implementing the CCurve Class	927
Exercising the CCurve Class	929
Creating the Sketch Document	929
Using a CTypedPtrList Template	929
Improving the View	935
Updating Multiple Views	935
Scrolling Views	937
Using MM_LOENGLISH Mapping Mode	941
Deleting and Moving Shapes	943
Implementing a Context Menu	943
Associating a Menu with a Class	945
Choosing a Context Menu	946
Highlighting Elements	952
Servicing the Menu Messages	956
Dealing with Masked Elements	964
Extending CLRSketcher	965
Coordinate System Transformations	965
Defining a Curve	967
Defining a Sketch Class	969
Drawing the Sketch in the Paint Event Handler	971
Implementing Element Highlighting	972
Creating Context Menus	976
Summary	981
Exercises	983
Chapter 17: Working with Dialogs and Controls	985
Understanding Dialogs	985
Understanding Controls	986
Common Controls	988
Creating a Dialog Resource	988
Adding Controls to a Dialog Box	988

Programming for a Dialog	990
Adding a Dialog Class	990
Modal and Modeless Dialogs	992
Displaying a Dialog	992
Supporting the Dialog Controls	995
Initializing the Controls	996
Handling Radio Button Messages	997
Completing Dialog Operations	998
Adding Pen Widths to the Document	998
Adding Pen Widths to the Elements	999
Creating Elements in the View	1000
Exercising the Dialog	1001
Using a Spin Button Control	1001
Adding the Scale Menu Item and Toolbar Button	1002
Creating the Spin Button	1003
Generating the Scale Dialog Class	1004
Displaying the Spin Button	1007
Using the Scale Factor	1008
Scaleable Mapping Modes	1008
Setting the Document Size	1009
Setting the Mapping Mode	1010
Implementing Scrolling with Scaling	1012
Working with Status Bars	1013
Adding a Status Bar to a Frame	1014
Using a List Box	1018
Removing the Scale Dialog	1018
Creating a List Box Control	1019
Using an Edit Box Control	1021
Creating an Edit Box Resource	1022
Creating the Dialog Class	1023
Adding the Text Menu Item	1025
Defining a Text Element	1026
Implementing the CText Class	1027
Creating a Text Element	1028
Dialogs and Controls in CLR Sketcher	1030
Adding a Dialog	1031
Creating Text Elements	1037
Summary	1044
Exercises	1045

Chapter 18: Storing and Printing Documents	1047
Understanding Serialization	1047
Serializing a Document	1048
Serialization in the Document Class Definition	1048
Serialization in the Document Class Implementation	1049
Functionality of CObject-Based Classes	1052
How Serialization Works	1053
How to Implement Serialization for a Class	1054
Applying Serialization	1055
Recording Document Changes	1055
Serializing the Document	1056
Serializing the Element Classes	1058
Exercising Serialization	1061
Moving Text	1062
Printing a Document	1064
The Printing Process	1065
Implementing Multipage Printing	1068
Getting the Overall Document Size	1068
Storing Print Data	1069
Preparing to Print	1070
Cleaning Up After Printing	1072
Preparing the Device Context	1072
Printing the Document	1073
Getting a Printout of the Document	1077
Serialization and Printing in CLR Sketcher	1078
Understanding Binary Serialization	1078
Serializing a Sketch	1083
Printing a Sketch	1089
Summary	1090
Exercises	1091
Chapter 19: Writing Your Own DLLs	1093
Understanding DLLs	1093
How DLLs Work	1095
Contents of a DLL	1098
DLL Varieties	1098
Deciding What to Put in a DLL	1099

Contents

Writing DLLs	1100
Writing and Using an Extension DLL	1100
Exporting Variables and Functions from a DLL	1108
Importing Symbols into a Program	1109
Implementing the Export of Symbols from a DLL	1109
Summary	1112
Exercises	1112
Chapter 20: Connecting to Data Sources	1113
Database Basics	1113
A Little SQL	1116
Retrieving Data Using SQL	1116
Joining Tables Using SQL	1118
Sorting Records	1120
Database Support in MFC	1120
MFC Classes Supporting ODBC	1121
Creating a Database Application	1122
Registering an ODBC Database	1122
Generating an MFC ODBC Program	1124
Understanding the Program Structure	1127
Exercising the Example	1138
Sorting a Recordset	1138
Modifying the Window Caption	1139
Using a Second Recordset Object	1140
Adding a Recordset Class	1141
Adding a View Class for the Recordset	1143
Customizing the Recordset	1147
Accessing Multiple Table Views	1150
Viewing Orders for a Product	1156
Viewing Customer Details	1156
Adding the Customer Recordset	1156
Creating the Customer Dialog Resource	1157
Creating the Customer View Class	1158
Adding a Filter	1160
Implementing the Filter Parameter	1162
Linking the Order Dialog to the Customer Dialog	1163
Exercising the Database Viewer	1165
Summary	1166
Exercises	1166

Chapter 21: Updating Data Sources	1167
Update Operations	1167
CRecordset Update Operations	1168
Transactions	1170
A Simple Update Example	1172
Customizing the Application	1173
Managing the Update Process	1175
Implementing Update Mode	1177
Adding Rows to a Table	1186
The Order Entry Process	1187
Creating the Resources	1187
Creating the Recordsets	1188
Creating the Recordset Views	1188
Adding Controls to the Dialog Resources	1193
Implementing Dialog Switching	1197
Creating an Order ID	1200
Storing the Order Data	1205
Selecting Products for an Order	1207
Adding a New Order	1209
Summary	1214
Exercises	1215
Chapter 22: More on Windows Forms Applications	1217
Creating the Application GUI	1217
Adding a Tab Control	1220
Using GroupBox Controls	1222
Using Button Controls	1224
Using the WebBrowser Control	1226
Operation of the Winning Application	1227
Adding a Context Menu	1228
Creating Event Handlers	1228
Handling Events for the Limits Menu	1235
Creating a Dialog Box	1235
Using the Dialog Box	1240
Adding the Second Dialog	1245
Implementing the Help > About Menu Item	1253
Handling a Button Click	1253
Responding to the Context Menu	1256
Summary	1263
Exercises	1264

Contents

Chapter 23: Accessing Data Sources in a Windows Forms Application	1265
Working with Data Sources	1266
Accessing and Displaying Data	1267
Using a DataGridView Control	1267
Using a DataGridView Control in Unbound Mode	1269
Customizing a DataGridView Control	1275
Customizing Header Cells	1276
Customizing Non-Header Cells	1277
Dynamically Setting Cell Styles	1284
Using Bound Mode	1290
The BindingSource Component	1290
Using the BindingNavigator Control	1296
Binding to Individual Controls	1299
Working with Multiple Tables	1302
Summary	1304
Exercises	1305
Appendix A: C++ Keywords	1307
Appendix B: ASCII Codes	1309
Appendix C: Windows Message Types	1317
Index	1319

Introduction

Welcome to *Beginning Visual C++® 2008*. With this book you can become an effective C++ programmer. The latest development system from Microsoft, Visual Studio 2008, supports two distinct but closely related flavors of the C++ language; it fully supports the original ISO/ANSI standard C++, and you also get support for a new version of C++ called C++/CLI that was developed by Microsoft and is now an ECMA standard. These two versions of C++ are complementary and fulfill different roles. ISO/ANSI C++ is there for the development of high-performance applications that run natively on your computer whereas C++/CLI has been developed specifically for writing applications that target the .NET Framework. This book will teach you how to write applications in both versions of C++.

You get quite a lot of assistance from automatically generated code when writing ISO/ANSI C++ programs, but you still need to write a lot of C++ yourself. You need a solid understanding of object-oriented programming techniques, as well as a good appreciation of what's involved in programming for Windows. Although C++/CLI targets the .NET Framework, it also is the vehicle for the development of Windows Forms applications that you can develop with little or in some cases no explicit code writing. Of course, when you do have to add code to a Windows Forms application, even though it may be a very small proportion of the total, you still need an in-depth knowledge of the C++/CLI language. ISO/ANSI C++ remains the language of choice for many professionals, but the speed of development that C++/CLI and Windows Forms applications bring to the table make that essential, too. For this reason I cover both flavors of C++ in this book.

Whom This Book Is For

This book is aimed at teaching you how to write C++ applications for the Microsoft Windows operating system using Visual C++ 2008 or any edition of Visual Studio 2008. I make no assumptions about prior knowledge of any particular programming language. This tutorial is for you if:

- ❑ You have a little experience programming in some other language, such as BASIC for example, and you are keen to learn C++ and develop practical Microsoft Windows programming skills.
- ❑ You have some experience in C or C++, but not in a Microsoft Windows context and want to extend your skills to program for the Windows environment using the latest tools and technologies.
- ❑ You have some knowledge of C++ and you want to extend your C++ skills to include C++/CLI.
- ❑ You are a newcomer to programming and sufficiently keen to jump in the deep end with C++. To be successful you need to have at least a rough idea of how your computer works, including the way in which the memory is organized and how data and instructions are stored.

What This Book Covers

My objective with this book is to teach you the essentials of C++ programming using both of the technologies supported by Visual C++ 2008. The book provides a detailed tutorial on both flavors of the C++ language, on native ISO/ANSI C++ Windows application development using the Microsoft Foundation Classes (MFC), and on the development of C++/CLI Windows applications using Windows Forms.

Because of the importance and pervasiveness of database technology today, the book also includes introductions to the techniques you can use for accessing data sources in both MFC and Windows Forms applications. MFC applications are relatively coding-intensive compared to Windows Forms applications. This is because you create the latter using a highly developed design capability in Visual C++ 2008 that enables you to assemble the entire graphical user interface (GUI) for an application graphically and have all the code that creates it generated automatically. For this reason, there are more pages in the book devoted to MFC programming than to Windows Forms programming.

How This Book Is Structured

The contents of this book are structured as follows:

- ❑ Chapter 1 introduces you to the basic concepts you need to understand for programming in C++ for native applications and for .NET Framework applications, together with the main ideas embodied in the Visual C++ 2008 development environment. It describes how you use the capabilities of Visual C++ 2008 for creating the various kinds of C++ applications you learn about in the rest of the book.
- ❑ Chapters 2 to 9 are dedicated to teaching you both versions of the C++ language. The content of each of the Chapters 2 through 9 is structured in a similar way; the first half of each chapter deals with ISO/ANSI C++ topics, and the second half deals with C++/CLI.
- ❑ Chapter 10 teaches you how you use the Standard Template Library (STL), which is a powerful and extensive set of tools for organizing and manipulating data in your native C++ programs. The STL is application-neutral so you will be able to apply it in a wide range of contexts. Chapter 10 also teaches you the STL/CLR, which is new in Visual C++ 2008. This is a version of the STL for C++/CLI applications.
- ❑ Chapter 11 introduces you to techniques for finding errors in your C++ programs.
- ❑ Chapter 12 discusses how Microsoft Windows applications are structured and describes and demonstrates the essential elements that are present in every Windows application. The chapter explains elementary examples of Windows applications using ISO/ANSI C++ and the Windows API and the MFC, as well as an example of a basic Windows Forms application in C++/CLI.
- ❑ Chapters 13 to 18 describe in detail the capabilities provided by the MFC for building a GUI and how you use the equivalent facilities in a program for the .NET Framework. You learn how you create and use common controls to build the graphical user interface for your application and how you handle the events that result from user interactions with your program. In the process, you create a substantial working application in native C++, and a program with essentially the same functionality in C++/CLI. In addition to the techniques you learn for building a GUI, the applications that you develop also show you how you print documents and how you save them on disk.

- ❑ Chapter 19 teaches you the essentials you need to know for creating your own libraries using MFC. You learn about the different kinds of libraries you can create, and you develop working examples of these that work with the application that you have evolved over the preceding six chapters.
- ❑ In Chapters 20 and 21, you learn about accessing data sources in an MFC application. You gain experience in accessing a database in read-only mode; then you learn the fundamental programming techniques for updating a database using MFC. The examples use the Northwind database that can be downloaded from the Web, but you can also apply the techniques described to your own data source.
- ❑ In Chapter 22 you work with Windows Forms and C++/CLI to build an example that teaches you how to create, customize, and use more Windows Forms controls in an application. You gain practical experience by building a second C++/CLI application incrementally throughout the chapter.
- ❑ Chapter 23 builds on the knowledge you gain in Chapter 22 and shows how the controls available for accessing data sources work, and how you customize them. You also learn how you can create an application for accessing a database with virtually no coding at all on your part.

All chapters include numerous working examples that demonstrate the programming techniques that are discussed. Every chapter concludes with a summary of the key points that were covered, and most chapters include a set of exercises at the end that you can attempt to apply what you have learned. Solutions to the exercises, together with all the code from the book, are available for download from the publisher's Web site (see the "Source Code" section later in this Introduction for more details).

The tutorial on the C++ language uses examples that are console programs with simple command-line input and output. This approach enables you to learn the various capabilities of C++ without getting bogged down in the complexities of Windows GUI programming. Programming for Windows is really only practicable after you have a thorough understanding of the programming language.

If you want to keep things as simple as possible, you can just learn ISO/ANSI C++ programming in the first instance. Each of the chapters that cover the C++ language (Chapters 2 to 9) first discusses particular aspects of the capabilities of ISO/ANSI C++, followed by the new features introduced by C++/CLI in the same context. The reason for organizing things this way is that C++/CLI is defined as an extension to the ISO/ANSI standard language, so an understanding of C++/CLI is predicated on knowledge of ISO/ANSI C++. Thus, you can just read the ISO/ANSI topics in each of Chapters 2 to 21 and ignore the C++/CLI sections that follow. You then can progress to Windows application development with ISO/ANSI C++ without having to keep the two versions of the language in mind. You can return to C++/CLI when you are comfortable with ISO/ANSI C++. Of course, you can also work straight through and add to your knowledge of both versions of the C++ language incrementally.

What You Need to Use This Book

To use this book you need any of Visual Studio 2008 Standard Edition, Visual Studio 2008 Professional Edition, or Visual Studio 2008 Team System. Note that Visual C++ Express 2008 is *not* sufficient because the MFC is not included.

Visual Studio 2008 requires Windows XP (x86 or x64) with Service Pack 2 or later, Windows Server 2003 with Service Pack 1 or later, or any edition of Windows Vista except Starter Edition. To install any of the

Introduction

three Visual Studio 2008 editions identified you need to have a 1.6 GHz processor with at least 384MB of memory (at least 768MB for Windows Vista) and at least 2.2GB of hard-disk space available. To install the full MSDN documentation that comes with the product you'll need an additional 1.8GB available on the installation drive.

The database examples in the book use the Northwind Traders database. You can find the download for this database by searching for "Northwind Traders" on <http://msdn.microsoft.com>. Of course, you can also adapt the examples to work with a database of your choice.

Most importantly, to get the most out of this book you need a willingness to learn, and a determination to master the most powerful programming tool for Windows applications presently available. You need the dedication to type in and work through all the examples and try out the exercises in the book. This sounds more difficult than it is, and I think you'll be surprised how much you can achieve in a relatively short period of time. Keep in mind that *everybody* who learns programming gets bogged down from time to time, but if you keep at it, things become clearer and you'll get there eventually. This book helps you to start experimenting on your own and, from there, to become a successful C++ programmer.

Using the Windows Classic Theme

If you're working in Windows Vista with Visual Studio 2008, you may have noticed that the view looks amazing. The transparency offered by the Aero Glass interface is quite breathtaking at first glance (and even many glances afterward). When you add in all of the visual effects that Vista has to offer, you might wonder why anyone would object to such a nice work environment. However, after a few hours of watching windows bursting forth and seeing the display dazzle your vision, you may prefer a setting that is less likely to cause eye fatigue. More importantly, you may notice a significant drop in your productivity because all of this eye candy robs your system of important processing cycles.

Eye candy is nice, but isn't it nicer to get home on time after a long day writing code? That's one reason why this book uses the Windows Classic theme to show Visual Studio 2008 windows. Another reason is that if you are still using Windows XP, the fancy Vista windows would not mean very much to you. The Windows Classic theme is common to both operating systems so it will fit with whatever operating system you are using, and it's definitely friendlier to your eyes than the Aero Glass interface.

If you are using Vista, I encourage you to try the various themes that Vista offers to see if they work for you. However, if you'd like to use the same theme in Vista as I have used for this book, then you can follow these steps to obtain it.

1. Right-click the Desktop and choose Personalize from the context menu. The Personalize window is shown in Figure I-1. This window provides access to all of the display settings you need to obtain the Windows Classic view.
2. Click Theme to display the Theme Settings dialog box shown in Figure I-2.
3. Choose Windows Classic in the Theme field and click OK. At this point, your display will begin looking very much like mine. Of course, you still have all of those special effects to consider. The next set of steps will get rid of the special effects.
4. Close the Personalize window.