

# *Ivor Horton's* Beginning **Visual C++\* 2012**

**Ivor Horton** 

INTRODUCTIO	N xxxv
CHAPTER 1	Programming with Visual C++1
CHAPTER 2	Data, Variables, and Calculations
CHAPTER 3	Decisions and Loops
CHAPTER 4	Arrays, Strings, and Pointers 135
CHAPTER 5	Introducing Structure into Your Programs
CHAPTER 6	More about Program Structure 231
CHAPTER 7	Defining Your Own Data Types 277
CHAPTER 8	More on Classes
CHAPTER 9	Class Inheritance and Virtual Functions
CHAPTER 10	The Standard Template Library 491
CHAPTER 11	Windows Programming Concepts
CHAPTER 12	Windows Programming with the Microsoft Foundation Classes (MFC)
CHAPTER 13	Working with Menus and Toolbars
CHAPTER 14	Drawing in a Window
CHAPTER 15	Improving the View
CHAPTER 16	Working with Dialogs and Controls
CHAPTER 17	Storing and Printing Documents
CHAPTER 18	Programming Windows 8 Apps849
INDEX	

## Visual C ++<sup>®</sup> 2012

IVOR HORTON'S BEGINNING

IVOR HORTON'S BEGINNING

## Visual C++® 2012

Ivor Horton



John Wiley & Sons, Inc.

#### Ivor Horton's Beginning Visual C++® 2012

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

Copyright © 2012 by Ivor Horton

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

Published simultaneously in Canada

ISBN: 978-1-118-36808-4 ISBN: 978-1-118-43941-8 (ebk) ISBN: 978-1-118-41703-4 (ebk) ISBN: 978-1-118-43431-4 (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: 2012946046

**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. 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, who for so many years has given me unconditional support and love in whatever I choose to do. I could not have written this without her.

#### **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 many 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 EDITOR



**MARC GREGOIRE** is a software engineer from Belgium. He graduated from the Catholic 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 big 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 big telecom operators. This required working in international teams stretching from South America and USA to EMEA and Asia. Now, Marc is working for Nikon Metrology on 3D scanning software.

His main expertise is C/C++, and specifically Microsoft VC++ and the MFC framework. Next to C/C++, Marc also likes C# and uses PHP for creating web pages. In addition to his main interest for Windows development, he also has experience in developing C++ programs running 24/7 on Linux platforms; for example, EIB home automation-controlling and monitoring software.

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) and an active member on the CodeGuru forum (as Marc G). He also creates freeware and shareware programs that are distributed through his website at www.nuonsoft.com, and maintains a blog on www.nuonsoft.com/blog/.

#### CREDITS

Executive Editor Robert Elliott

Project Editor Ed Connor

**Technical Editor** Marc Gregoire

**Production Editor** Daniel Scribner

Copy Editor Kim Cofer

Editorial Manager Mary Beth Wakefield

Freelancer Editorial Manager Rosemarie Graham

Associate Director of Marketing David Mayhew

Marketing Manager Ashley Zurcher

Business Manager Amy Knies Production Manager Tim Tate

Vice President and Executive Group Publisher Richard Swadley

Vice President and Executive Publisher Neil Edde

Associate Publisher Jim Minatel

**Project Coordinator, Cover** Katie Crocker

**Proofreaders** James Saturnio, Word One Sarah Kaikini, Word One

Indexer Ron Strauss

**Cover Designer** Ryan Sneed

**Cover Image** © Kyu Oh / iStockPhoto

## 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 editor, Marc Gregoire, for doing such a fantastic job of reviewing the text and checking out all the code fragments and examples. He has an uncanny knack for finding my errors, and his many constructive comments and suggestions have undoubt-edly made the book a much better tutorial.

### CONTENTS

INTRODUCTION	XXX
CHAPTER 1: PROGRAMMING WITH VISUAL C++	
Learning with Visual C++	
Writing C++ Applications	2
Learning Desktop Applications Programming	Э
Learning C++	3
Console Applications	Z
Windows Programming Concepts	Z
What Is the Integrated Development Environment?	5
The Editor	6
The Compiler	6
The Linker	6
The Libraries	6
Using the IDE	7
Toolbar Options	8
Dockable Toolbars	g
Documentation	10
Projects and Solutions	10
Defining a Project	10
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	3.
The #include Directive — Header Files	32
Namespaces and the Using Declaration	33
The main() Function	34
Program Statements	34

Whitespace	37
Statement Blocks	37
Automatically Generated Console Programs	37
Precompiled Header Files	38
Main Function Names	39
Defining Variables	39
Naming Variables	39
Keywords in C++	40
Declaring Variables	41
Initial Values for Variables	41
Fundamental Data Types	42
Integer Variables	42
Character Data Types	44
Integer Type Modifiers	45
The Boolean Type	46
Floating-Point Types	47
Fundamental Types in C++	47
Literals	48
Defining Synonyms for Data Types	49
Basic Input/Output Operations	50
Input from the Keyboard	50
Output to the Command Line	50
Formatting the Output	51
Escape Sequences	53
Calculating in C++	55
The Assignment Statement	55
Arithmetic Operations	55
The const Modifier	58
Constant Expressions	58
Program Input	59
Calculating the Result	59
Displaying the Result	60
Calculating a Remainder	61
Modifying a Variable	61
The Increment and Decrement Operators	62
The Sequence of Calculation	65
Operator Precedence	65
Type Conversion and Casting	66
Type Conversion in Assignments	67
Explicit Type Conversion	68
Old-Style Casts	69

The auto Keyword	69
Discovering Types	70
The Bitwise Operators	70
The Bitwise AND	71
The Bitwise OR	72
The Bitwise Exclusive OR	73
The Bitwise NOT	74
The Bitwise Shift Operators	74
Introducing Lvalues and Rvalues	76
Understanding Storage Duration and Scope	77
Automatic Variables	78
Positioning Variable Declarations	80
Global Variables	80
Static Variables	84
Variables with Specific Sets of Values	85
Old Enumerations	85
Type-Safe Enumerations	87
Namespaces	90
Declaring a Namespace	91
Multiple Namespaces	92
Summary	93
Sammary	93
CHAPTER 3: DECISIONS AND LOOPS	97
CHAPTER 3: DECISIONS AND LOOPS	
	97
CHAPTER 3: DECISIONS AND LOOPS Comparing Values	97 97
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement	<b>97</b> <b>97</b> 99
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements	<b>97</b> <b>97</b> 99 100
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement	<b>97</b> 99 100 102
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements	<b>97</b> 99 100 102 104
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions	<b>97</b> 99 100 102 104 106
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions Logical AND	<b>97</b> 99 100 102 104 106 107
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions Logical AND Logical OR	<b>97</b> 99 100 102 104 106 107 107
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions Logical AND Logical OR Logical NOT	<b>97</b> 99 100 102 104 106 107 107
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions Logical AND Logical OR Logical NOT The Conditional Operator	97 99 100 102 104 106 107 107 108 109
CHAPTER 3: DECISIONS AND LOOPS  Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions Logical OR Logical OR Logical NOT The Conditional Operator The switch Statement	<b>97</b> 99 100 102 104 106 107 107 107 108 109 111
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions Logical AND Logical OR Logical NOT The Conditional Operator The switch Statement Unconditional Branching	97 99 100 102 104 106 107 107 107 108 109 111 115
CHAPTER 3: DECISIONS AND LOOPS Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions Logical AND Logical OR Logical NOT The Conditional Operator The switch Statement Unconditional Branching Repeating a Block of Statements	97 99 100 102 104 106 107 107 107 108 109 111 115 <b>115</b>
CHAPTER 3: DECISIONS AND LOOPS  Comparing Values The if Statement Nested if Statements The Extended if Statement Nested if-else Statements Logical Operators and Expressions Logical AND Logical OR Logical NOT The Conditional Operator The switch Statement Unconditional Branching Repeating a Block of Statements What Is a Loop?	97 99 100 102 104 106 107 107 107 108 109 111 115 <b>115</b> 116

The while Loop	126
The do-while Loop	128
The Range-Based for Loop	129
Nested Loops	130
Summary	133
CHAPTER 4: ARRAYS, STRINGS, AND POINTERS	135
Handling Multiple Data Values	405
of the Same Type	135
Arrays	136
Declaring Arrays	137
Initializing Arrays	140
Using the Range-Based for Loop	142
Character Arrays and String Handling	143
String Input	144
Using the Range-Based for Loop with Strings	146
Multidimensional Arrays	146
Initializing Multidimensional Arrays	147
Indirect Data Access	150
What Is a Pointer?	150
Declaring Pointers	150
The Address-Of Operator	151
Using Pointers	152
The Indirection Operator	152
Why Use Pointers?	152
Initializing Pointers	152
Pointers to char	155
The sizeof Operator	159
Constant Pointers and Pointers to Constants	161
Pointers and Arrays	163
Pointer Arithmetic	164
Using Pointers with Multidimensional Arrays	168
Pointer Notation with Multidimensional Arrays	169
Dynamic Memory Allocation	170
The Free Store, Alias the Heap	170
The new and delete Operators	171
Allocating Memory Dynamically for Arrays	172
Dynamic Allocation of Multidimensional Arrays	175
Using References	176
What Is a Reference?	176
Declaring and Initializing Lvalue References	176

Using References in a Range-Based for Loop	177
Rvalue References	178
Library Functions for Strings	178
Finding the Length of a Null-Terminated String	179
Joining Null-Terminated Strings	179
Copying Null-Terminated Strings	181
Comparing Null-Terminated Strings	182
Searching Null-Terminated Strings	183
Summary	185
CHAPTER 5: INTRODUCING STRUCTURE INTO YOUR PROGRAMS	189
Understanding Functions	189
Why Do You Need Functions?	191
Structure of a Function	191
The Function Header	191
The Function Body	193
The return Statement	193
Alternative Function Syntax	194
Using a Function	194
Function Prototypes	194
Passing Arguments to a Function	198
The Pass-by-Value Mechanism	199
Pointers as Arguments to a Function	200
Passing Arrays to a Function	202
Passing Multidimensional Arrays to a Function	204
References as Arguments to a Function	206
Use of the const Modifier	208
Rvalue Reference Parameters	210
Arguments to main( )	212
Accepting a Variable Number of Function Arguments	214
Returning Values from a Function	216
Returning a Pointer	216
A Cast-Iron Rule for Returning Addresses	218
Returning a Reference	219
A Teflon-Coated Rule: Returning References	222
Static Variables in a Function	222
Recursive Function Calls	224
Using Recursion	227
Summary	227

CHAPTER 6: MORE ABOUT PROGRAM STRUCTURE	231
Pointers to Functions	231
Declaring Pointers to Functions	232
A Pointer to a Function as an Argument	235
Arrays of Pointers to Functions	237
Initializing Function Parameters	238
Exceptions	239
Throwing Exceptions	241
Catching Exceptions	242
Rethrowing Exceptions	244
Exception Handling in the MFC	244
Handling Memory Allocation Errors	245
Function Overloading	247
What Is Function Overloading?	247
Reference Types and Overload Selection	250
When to Overload Functions	251
Function Templates	251
Using a Function Template	251
Using the decltype Operator	254
An Example Using Functions	256
Implementing a Calculator	257
Analyzing the Problem	257
Eliminating Blanks from a String	260
How the Function Functions	260
Evaluating an Expression	260
How the Function Functions	262
Getting the Value of a Term	263
How the Function Functions	264
Analyzing a Number	264
How the Function Functions	266
Putting the Program Together	268
How the Function Functions	269
Extending the Program	269
How the Function Functions	271
Extracting a Substring	271
How the Function Functions	273
Running the Modified Program	273
Summary	274

#### CONTENTS

CHAPTER 7: DEFINING YOUR OWN DATA TYPES	277
The struct in C++	277
What Is a struct?	278
Defining a struct	278
Initializing a struct	279
Accessing the Members of a struct	279
IntelliSense Assistance with Structures	283
The struct RECT	285
Using Pointers with a struct	285
Accessing Structure Members through a Pointer	286
The Indirect Member Selection Operator	287
Types, Objects, Classes, and Instances	287
First Class	289
Operations on Classes	289
Terminology	290
Understanding Classes	290
Defining a Class	291
Access Control in a Class	291
Declaring Objects of a Class	291
Accessing the Data Members of a Class	292
Member Functions of a Class	294
Positioning a Member Function Definition	296
Inline Functions	297
Class Constructors	298
What Is a Constructor?	298
The Default Constructor	300
Default Parameter Values	303
Using an Initialization List in a Constructor	305
Making a Constructor Explicit	306
Private Members of a Class	307
Accessing private Class Members	310
The friend Functions of a Class	310
Placing friend Function Definitions Inside the Class	312
The Default Copy Constructor	313
The Pointer this	314
const Objects	317
const Member Functions of a Class	318
Member Function Definitions Outside the Class	319
Arrays of Objects	320

Static Members of a Class	322
Static Data Members	322
Static Function Members of a Class	325
Pointers and References to Objects	325
Pointers to Objects	326
References to Class Objects	328
Implementing a Copy Constructor	329
Summary	330
CHAPTER 8: MORE ON CLASSES	333
Class Destructors	334
What Is a Destructor?	334
The Default Destructor	334
Destructors and Dynamic Memory Allocation	337
Implementing a Copy Constructor	340
Sharing Memory Between Variables	342
Defining Unions	343
Anonymous Unions	344
Unions in Classes and Structures	345
Operator Overloading	345
Implementing an Overloaded Operator	346
Implementing Full Support for Comparison Operators	350
Overloading the Assignment Operator	354
Fixing the Problem	355
Overloading the Addition Operator	359
Overloading the Increment and Decrement Operators	364
Overloading the Function Call Operator	365
The Object Copying Problem	366
Avoiding Unnecessary Copy Operations	367
Applying Rvalue Reference Parameters	370
Named Objects are Lvalues	372
Default Class Members	377
Class Templates	378
Defining a Class Template	379
Template Member Functions	381
Creating Objects from a Class Template	382
Class Templates with Multiple Parameters	386
Templates for Function Objects	388
Perfect Forwarding	388
Using Classes	392
The Idea of a Class Interface	392
Defining the Problem	393

Implementing the CBox Class	393
Comparing CBox Objects	395
Combining CBox Objects	396
Analyzing CBox Objects	398
Organizing Your Program Code	412
Naming Program Files	413
Library Classes for Strings	414
Creating String Objects	414
Concatenating Strings	416
Accessing and Modifying Strings	420
Comparing Strings	424
Searching Strings	428
Summary	437
CHAPTER 9: CLASS INHERITANCE AND VIRTUAL FUNCTIONS	441
Object-Oriented Programming Basics	441
Inheritance in Classes	443
What Is a Base Class?	443
Deriving Classes from a Base Class	444
Access Control Under Inheritance	447
Constructor Operation in a Derived Class	450
Declaring Protected Class Members	454
The Access Level of Inherited Class Members	457
The Copy Constructor in a Derived Class	458
Preventing Class Derivation	461
Class Members as Friends	462
Friend Classes	464
Limitations on Class Friendship	464
Virtual Functions	464
What Is a Virtual Function?	467
Ensuring Correct Virtual Function Operation	469
Preventing Function Overriding	469
Using Pointers to Class Objects	470
Using References with Virtual Functions	472
Incomplete Class Declaration	473
Pure Virtual Functions	473
Abstract Classes	474
Indirect Base Classes	477
Virtual Destructors	479
Casting Between Class Types	483
Nested Classes	483
Summary	487

CHAPTER 10: THE STANDARD TEMPLATE LIBRARY	491
What Is the Standard Template Library?	491
Containers	492
Allocators	494
Comparators	494
Container Adapters	495
Iterators	495
Iterator Categories	495
SCARY Iterators	497
std::begin( ) and std::end( ) Functions	497
Smart Pointers	497
Using unique_ptr Objects	498
Using shared_ptr Objects	499
Accessing the Raw Pointer in a Smart Pointer	500
Casting SmartPointers	500
Algorithms	500
Function Objects in the STL	501
Function Adapters	502
The Range of STL Containers	502
Sequence Containers	502
Creating Vector Containers	504
The Capacity and Size of a Vector Container	507
Accessing the Elements in a Vector	512
Inserting and Deleting Elements in a Vector	513
Insert Operations	513
Emplace Operations	514
Erase Operations	515
Swap and Assign Operations	515
Storing Class Objects in a Vector	516
Sorting Vector Elements	522
Storing Pointers in a Vector	523
Array Containers	526
Double-Ended Queue Containers	529
Using List Containers	533
Adding Elements to a List	533
Accessing Elements in a List	535
Sorting List Elements	535
Other Operations on Lists	538
Using forward_list Containers	544
Using Other Sequence Containers	546
Sorting List Elements Other Operations on Lists Using forward_list Containers	53 53 54

Queue Containers	546
Priority Queue Containers	549
Stack Containers	555
The tuple Class Template	557
Associative Containers	561
Using Map Containers	561
Storing Objects	562
Accessing Objects	564
Other Map Operations	565
Using a Multimap Container	574
More on Iterators	575
Using Input Stream Iterators	575
Using Inserter Iterators	578
Using Output Stream Iterators	580
More on Function Objects	582
More on Algorithms	584
Type Traits and Static Assertions	587
Lambda Expressions	588
The Capture Clause	589
Capturing Specific Variables	590
Templates and Lambda Expressions	591
Naming a Lambda Expression	595
Summary	598
CHAPTER 11: WINDOWS PROGRAMMING CONCEPTS	601
Windows Programming Basics	602
Elements of a Window	602
Windows Programs and the Operating System	604
Event-Driven Programs	605
Windows Messages	605
The Windows API	605
Windows Data Types	606
Notation in Windows Programs	607
The Structure of a Windows Program	608
The WinMain() Function	609
Specifying a Program Window	611
Creating a Program Window	613
Initializing the Program Window	615
Dealing with Windows Messages	616
A Complete WinMain() Function	620
How It Works	621

Processing Windows Messages	621
The WindowProc() Function	622
Decoding a Windows Message	622
Ending the Program	625
A Complete WindowProc() Function	625
How It Works	626
The Microsoft Foundation Classes	627
MFC Notation	627
How an MFC Program Is Structured	628
Summary	632
CHAPTER 12: WINDOWS PROGRAMMING WITH THE MICROSOFT FOUNDATION	605
CLASSES (MFC)	635
The MFC Document/View Concept	636
What Is a Document?	636
Document Interfaces	636
What Is a View?	636
Linking a Document and Its Views	637
Document Templates	638
Document Template Classes	638
Your Application and MFC	639
Creating MFC Applications	640
Creating an SDI Application	641
MFC Application Wizard Output	645
Viewing Project Files	647
Viewing Classes	647
The Class Definitions	648
Creating an Executable Module	653
Running the Program	653
How the Program Works	654
Creating an MDI Application	655
Running the Program	656
Summary	657
CHAPTER 13: WORKING WITH MENUS AND TOOLBARS	659
Communicating with Windows	659
Understanding Message Maps	660
Message Handler Definitions	661
Message Categories	662
Handling Messages in Your Program	663
How Command Messages Are Processed	664