SIXTH EDITION

Microsoft® Visual Basic® 2015 *RELOADED*

Diane Zak

MICROSOFT®
VISUAL BASIC® 2015:
RELOADED

SIXTH EDITION

MICROSOFT® VISUAL BASIC® 2015: RELOADED

DIANE ZAK

CENGAGE Learning®

Australia • Brazil • Mexico • Singapore • United Kingdom • United States

Microsoft® Visual Basic® 2015: RELOADED, Sixth Edition
Diane Zak

Product Director: Kathleen McMahon

Product Team Manager: Kristin McNary

Associate Product Manager: Megan Chrisman

Senior Content Developer: Alyssa Pratt

Product Assistant: Abigail Pufpaff

Marketing Manager: Eric LaScola

Senior Production Director: Wendy Troeger

Production Director: Patty Stephan

Senior Content Project Manager: Jennifer K. Feltri-George

Managing Art Director: Jack Pendleton

Cover image(s): © Rudchenko Liliia/Shutterstock.com

Unless otherwise noted all screenshots are courtesy of Microsoft Corporation

Open Clip art source: OpenClipArto.org

For product information and technology assistance, contact us at
Cengage Learning Customer & Sales Support, 1-800-354-9706

For permission to use material from this text or product, submit all requests online at **www.cengage.com/permissions**
Further permissions questions can be emailed to
permissionrequest@cengage.com

Library of Congress Control Number: 2015947713
ISBN: 978-1-285-86019-0

Cengage Learning
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with employees residing in nearly 40 different countries and sales in more than 125 countries around the world. Find your local representative at **www.cengage.com**

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

For your course and learning solutions, visit **www.cengage.com**

Purchase any of our products at your local college store or at our preferred online store **www.cengagebrain.com**

Microsoft and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries. Cengage Learning is an independent entity from Microsoft Corporation, and not affiliated with Microsoft in any manner.

Notice to the Reader

Publisher does not warrant or guarantee any of the products described herein or perform any independent analysis in connection with any of the product information contained herein. Publisher does not assume, and expressly disclaims, any obligation to obtain and include information other than that provided to it by the manufacturer. The reader is expressly warned to consider and adopt all safety precautions that might be indicated by the activities described herein and to avoid all potential hazards. By following the instructions contained herein, the reader willingly assumes all risks in connection with such instructions. The publisher makes no representations or warranties of any kind, including but not limited to, the warranties of fitness for particular purpose or merchantability, nor are any such representations implied with respect to the material set forth herein, and the publisher takes no responsibility with respect to such material. The publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or part, from the readers' use of, or reliance upon, this material.

Printed in the United States of America
Print Number: 01 Print Year: 2016

Brief Contents

	Preface	xiv
	Read This Before You Begin	xviii
OVERVIEW	An Introduction to Programming	1
CHAPTER 1	An Introduction to Visual Basic 2015	7
CHAPTER 2	Creating a User Interface	73
CHAPTER 3	Memory Locations and Calculations	111
CHAPTER 4	Making Decisions in a Program	175
CHAPTER 5	More on the Selection Structure	229
CHAPTER 6	Repeating Program Instructions	285
CHAPTER 7	More on the Repetition Structure	355
CHAPTER 8	Sub and Function Procedures	405
CHAPTER 9	Arrays	455
CHAPTER 10	String Manipulation and Menus	505
CHAPTER 11	Structures and Sequential Files	549
CHAPTER 12	Access Databases and LINQ	603
CHAPTER 13	Creating Simple Web Applications	657
CHAPTER 14	Creating Classes and Objects	697
APPENDIX A	Answers to Mini-Quizzes	745
APPENDIX B	How To Boxes	757
APPENDIX C	Most Commonly Used Properties of Objects	761
APPENDIX D	Visual Basic Conversion Functions	765
APPENDIX E	Finding and Fixing Program Errors	767
	Index	783

Contents

Preface xiv

Read This Before You Begin xviii

OVERVIEW

An Introduction to Programming 1

Programming a Computer 2

The Programmer's Job 2

Employment Opportunities 2

Using the Chapters Effectively 3

Overview Summary 5

Key Terms 5

CHAPTER 1

An Introduction to Visual Basic 2015 7

Visual Basic 2015 8

Starting and Configuring Visual Studio 8

Creating a Visual Basic Windows Forms Application 11

Managing the Windows in the IDE 12

The Windows Form Designer Window 13

The Solution Explorer Window 14

The Properties Window 15

Properties of a Windows Form 15

Name and Text Properties 16

StartPosition Property 17

Font Property 17

Size Property 17

The Toolbox Window 18

The Label Control 20

The Button Control 21

The Picture Box Control 21

Using the Format Menu 21

The Code Editor Window 23

The `Me.Close()` Instruction 25

Saving a Solution 25

Starting and Ending an Application 26

Assigning a Value to a Property During Run Time 28

Printing the Code and User Interface 30

Closing the Current Solution 31

Opening an Existing Solution 31

Syntax Errors 32

Programming Tutorial 1 34

Programming Tutorial 2 . . . 53
Programming Example . . . 61
Chapter Summary . . . 63
Key Terms . . . 65
Review Questions . . . 67
Exercises . . . 68
Case Projects . . . 70

CHAPTER 2 Creating a User Interface . . . 73

Planning an Application . . . 74
Say Cheese! Company . . . 74
  Identifying the Application's Tasks . . . 74
  Identifying the Objects. . . . 76
  Identifying the Events . . . 77
Designing the User Interface . . . 79
  Control Placement . . . 80
  Graphics, Fonts, and Color . . . 82
Assigning Access Keys . . . 83
Controlling the Tab Order . . . 84
Designating the Default Button on a Form . . . 87
Printing an Interface from Code . . . 87
Playing Audio Files . . . 89
Programming Tutorial 1 . . . 90
Programming Tutorial 2 . . . 96
Programming Example . . . 98
Chapter Summary . . . 102
Key Terms . . . 103
Review Questions . . . 104
Exercises . . . 105
Case Projects . . . 108

CHAPTER 3 Memory Locations and Calculations . . . 111

Internal Memory . . . 112
Variables . . . 112
  Declaring a Variable in Code . . . 115
Assigning Data to an Existing Variable . . . 116
  Using the TryParse Method . . . 118
  Using the Convert Class Methods . . . 119
Arithmetic Expressions . . . 121
  Arithmetic Assignment Operators . . . 124
Scope and Lifetime . . . 125
  Variables with Procedure Scope . . . 126
  Variables with Class Scope . . . 129
  Static Variables . . . 130
Named Constants . . . 132
Option Statements . . . 133
  Option Explicit and Option Infer . . . 134
  Option Strict . . . 134

Completing the Say Cheese! Company's Application 137
Using Pseudocode to Plan a Procedure 138
Using a Flowchart to Plan a Procedure 139
Coding the calcButton's Click Event Procedure 141
Formatting Numeric Output 143
Coding the clearButton's Click Event Procedure 145
Coding the exitButton_Click and printButton_Click Procedures 146
Testing and Debugging the Application 148
Programming Tutorial 1 150
Programming Tutorial 2 156
Programming Example 159
Chapter Summary 162
Key Terms 164
Review Questions 166
Exercises 167
Case Projects 171

CHAPTER 4 Making Decisions in a Program 175

The Selection Structure 176
Single-Alternative and Dual-Alternative Selection Structures 176
The If...Then...Else Statement 182
Comparison Operators 183
Checking the Denominator in an Expression 185
Swapping Numeric Values. 186
String Concatenation. 189
Comparing Strings. 190
The Square Calculations Application 192
Comparing Boolean Values 195
Logical Operators 197
Using the Truth Tables 199
Summary of Operators 201
Generating Random Integers 202
Programming Tutorial 1 204
Programming Tutorial 2 210
Programming Example 214
Chapter Summary 217
Key Terms 218
Review Questions 220
Exercises 221
Case Projects 226

CHAPTER 5 More on the Selection Structure 229

Making More than One Decision 230
The Cookie Palace Application 232
Multiple-Alternative Selection Structures 238
The Select Case Statement 240
Specifying a Range of Values in a Case Clause 242
Using Radio Buttons in an Interface 243

The MessageBox.Show Method 246
Using the KeyPress Event 249
Programming Tutorial 1 251
Programming Tutorial 2 259
Programming Example 264
Chapter Summary 267
Key Terms 268
Review Questions 269
Exercises 272
Case Projects 280

CHAPTER 6 Repeating Program Instructions 285

The Repetition Structure 286
The Projected Sales Application 287
The Do...Loop Statement 290
Flowcharting a Loop 291
Coding the Projected Sales Application 292
Counters and Accumulators 295
The InputBox Function 297
Average Seminar Registrants Application 298
Including a List Box in an Interface 302
Adding Items to a List Box 302
Clearing the Items from a List Box 306
The Sorted Property 306
Accessing Items in a List Box 307
Determining the Number of Items in a List Box 307
The SelectedItem and SelectedIndex Properties 308
The SelectedValueChanged and SelectedIndexChanged Events 309
The Product Finder Application 311
The Color Viewer Application 314
Programming Tutorial 1 316
Programming Tutorial 2 326
Programming Example 334
Chapter Summary 337
Key Terms 338
Review Questions 340
Exercises 342
Case Projects 351

CHAPTER 7 More on the Repetition Structure 355

Counter-Controlled Loops 356
The For...Next Statement 356
The Projected Sales Application 359
Comparing the For...Next and Do...Loop Statements 361
Nested Repetition Structures 362
The Financial.Pmt Method 365
The Monthly Payment Application 367
Selecting the Existing Text in a Text Box 368

Coding the TextChanged Event Procedure 370
Coding the SelectedValueChanged and SelectedIndexChanged Event Procedures 371
Including a Combo Box in an Interface 372
Using an Image List Control 375
Programming Tutorial 1 378
Programming Tutorial 2 384
Programming Example 388
Chapter Summary 392
Key Terms 393
Review Questions 394
Exercises 396
Case Projects 401

CHAPTER 8 Sub and Function Procedures 405

Sub Procedures 406
The Gillian Company Application 408
Passing Variables 409
Passing Variables by Value 410
Passing Variables by Reference 411
Associating a Procedure with Different Objects and Events 415
Function Procedures 416
Using a Function in the Gillian Company Application 418
Converting Object Variables 421
Rounding Numbers 422
Programming Tutorial 1 423
Programming Tutorial 2 430
Programming Example 439
Chapter Summary 442
Key Terms 443
Review Questions 444
Exercises 446
Case Projects 450

CHAPTER 9 Arrays 455

Arrays 456
One-Dimensional Arrays 456
Declaring a One-Dimensional Array 457
Storing Data in a One-Dimensional Array 459
Determining the Number of Elements in an Array 459
Determining the Highest Subscript in a One-Dimensional Array 460
Traversing a One-Dimensional Array 461
The For Each...Next Statement 462
Calculating the Total and Average Values 462
Finding the Highest Value 464
Arrays and Collections 466
Accumulator and Counter Arrays 468
Sorting a One-Dimensional Array 469

Two-Dimensional Arrays . . . 471
Traversing a Two-Dimensional Array . . . 474
Totaling the Values Stored in a Two-Dimensional Array . . . 476
Searching a Two-Dimensional Array . . . 477
Programming Tutorial 1 . . . 480
Programming Tutorial 2 . . . 485
Programming Example . . . 488
Chapter Summary . . . 491
Key Terms . . . 492
Review Questions . . . 493
Exercises . . . 496
Case Projects . . . 501

CHAPTER 10 String Manipulation and Menus . . . 505

Working with Strings . . . 506
Determining the Number of Characters in a String . . . 506
Removing Characters from a String . . . 506
Inserting Characters in a String . . . 507
Aligning the Characters in a String . . . 509
Searching a String . . . 510
Accessing the Characters in a String . . . 511
Using Pattern Matching to Compare Strings . . . 513
Adding a Menu to a Form . . . 515
Programming Tutorial 1 . . . 517
Programming Tutorial 2 . . . 530
Programming Example . . . 535
Chapter Summary . . . 538
Key Terms . . . 539
Review Questions . . . 540
Exercises . . . 541
Case Projects . . . 546

CHAPTER 11 Structures and Sequential Files . . . 549

Structures . . . 550
Declaring a Using a Structure Variable . . . 550
Passing a Structure Variable to a Procedure . . . 552
Creating an Array of Structure Variables . . . 555
Sequential Access Files . . . 557
Writing Data to a Sequential Access File . . . 557
Closing an Output Sequential Access File . . . 559
Reading Data from a Sequential Access File . . . 560
Closing an Input Sequential Access File . . . 562
The FormClosing Event . . . 563
The Kettleson Club Application . . . 564
Aligning Columns of Information . . . 567
Writing and Reading Records . . . 569
Programming Tutorial 1 . . . 571
Programming Tutorial 2 . . . 577

Programming Example 586
Chapter Summary 589
Key Terms 590
Review Questions 592
Exercises 593
Case Projects 600

CHAPTER 12 Access Databases and LINQ 603

Database Terminology 604
Connecting an Application to a Microsoft Access Database 606
Binding the Objects in a Dataset 608
Having the Computer Create a Bound Control 609
The DataGridView Control 611
Visual Basic Code 613
Handling Errors in the Code 614
The Copy to Output Directory Property 615
Binding to an Existing Control 616
Coding the Next Record and Previous Record Buttons 618
Accessing the Value Stored in a Field 619
Creating a Query 620
Customizing a BindingNavigator Control 622
Using the LINQ Aggregate Operators 624
Programming Tutorial 1 627
Programming Tutorial 2 636
Programming Example 643
Chapter Summary 646
Key Terms 647
Review Questions 648
Exercises 650
Case Projects 654

CHAPTER 13 Creating Simple Web Applications 657

Web Site Applications 658
Creating a Web Site Application 660
Adding a Web Page to an Application 661
Customizing a Web Page 663
Coding the Controls on a Web Page 664
Starting a Web Application 664
Closing and Opening an Existing Web Application 665
Programming Tutorial 1 666
Programming Tutorial 2 679
Programming Example 687
Chapter Summary 688
Key Terms 689
Review Questions 689
Exercises 690
Case Projects 693

CHAPTER 14 Creating Classes and Objects 697
Object-Oriented Programming Terminology 698
Creating a Class 698
Example 1—A Class That Contains Public Variables Only 700
Private Variables and Public Property Procedures 702
Constructors 705
Methods Other Than Constructors 706
Example 2—A Class That Contains Private Variables,
Public Properties, and Methods 707
Example 3—Reusing a Class 710
Example 4—A Class That Contains a ReadOnly Property 711
Example 5—A Class That Contains Auto-Implemented Properties 713
Example 6—A Class That Contains Overloaded Methods 715
Programming Tutorial 1 719
Programming Tutorial 2 724
Programming Example 728
Chapter Summary 732
Key Terms 733
Review Questions 734
Exercises 736
Case Projects 741

APPENDIX A Answers to Mini-Quizzes 745

APPENDIX B How To Boxes 757

APPENDIX C Most Commonly Used Properties of Objects 761

APPENDIX D Visual Basic Conversion Functions 765

APPENDIX E Finding and Fixing Program Errors 767

Index 783

Preface

Microsoft Visual Basic 2015: RELOADED, Sixth Edition uses Visual Basic 2015, an object-oriented language, to teach programming concepts. This book is designed for a beginning programming course; however, it assumes students are familiar with basic Windows skills and file management.

Organization and Coverage

Microsoft Visual Basic 2015: RELOADED, Sixth Edition contains 14 chapters and five appendices (A through E). In the chapters, students with no previous programming experience learn how to plan and create their own interactive Windows applications. By the end of the book, students will have learned how to use TOE charts, pseudocode, and flowcharts to plan an application. They will also learn how to work with objects and write Visual Basic statements such as If...Then...Else, Select Case, Do...Loop, For...Next, and For Each...Next. Students will also learn how to manipulate variables, constants, strings, sequential access files, structures, and arrays. In Chapter 12, they will learn how to connect an application to a Microsoft Access database, and then use Language Integrated Query (LINQ) to query the database. Chapter 13 shows students how to create simple Web applications, and Chapter 14 shows them how to create their own classes and objects.

Approach

Like the previous editions, *Microsoft Visual Basic 2015: RELOADED, Sixth Edition* is distinguished from other textbooks because of its unique approach, which motivates students by demonstrating why they need to learn the concepts and skills presented. Each chapter begins with an introduction to one or more programming concepts. The concepts are illustrated with code examples and sample programs. The sample programs are provided to students to allow them to observe how the current concept can be utilized before they are introduced to the next concept. Following the concept portion in each chapter are two Programming Tutorials. Each Programming Tutorial guides students through the process of creating an application using the concepts covered in the chapter. A Programming Example follows the Programming Tutorials in each chapter. The Programming Example contains a completed application that demonstrates the chapter concepts. Following the Programming Example are the Summary, Key Terms, Review Questions, Exercises, and Case Projects sections.

Features

Microsoft Visual Basic 2015: RELOADED, Sixth Edition is an exceptional textbook because it also includes the following features:

READ THIS BEFORE YOU BEGIN This section is consistent with Cengage Learning's unequaled commitment to helping instructors introduce technology into the classroom. Technical considerations and assumptions about hardware, software, and default settings are listed in one place to help instructors save time and eliminate unnecessary aggravation.

DESIGNED FOR THE DIFFERENT LEARNING STYLES The three most common learning styles are visual, auditory, and kinesthetic. This book contains videos for visual and auditory learners, and Try It! files for kinesthetic learners.

OBJECTIVES The Review Questions, Exercises, and Case Projects are associated with one or more of the objectives listed at the beginning of the chapter.

VIDEOS These notes direct students to videos that explain and/or demonstrate one or more of the chapter's concepts, provide additional information about the concepts, or cover topics related to the concepts. The videos are available at *www.cengagebrain.com*. Search for the ISBN of your title (from the back cover of your book) using the search box at the top of the page. This will take you to the product page where free companion resources can be found.

TRY IT! FILES Each chapter has accompanying Try It! files that allow the student to practice a concept before moving on to the next concept.

MINI-QUIZZES Mini-Quizzes are strategically placed to test students' knowledge at various points in the chapter. Answers to the quiz questions are provided in Appendix A, allowing students to determine whether they have mastered the material covered thus far before continuing with the chapter.

HOW TO BOXES The How To boxes in each chapter summarize important concepts and provide a quick reference for students. The How To boxes that introduce new statements, functions, or methods contain the syntax and examples of using the syntax. Many of the How To boxes contain the steps for performing common tasks.

TIP These notes provide additional information about the current concept. Examples include alternative ways of writing statements or performing tasks, as well as warnings about common mistakes made when using a particular command and reminders of related concepts learned in previous chapters.

PROGRAMMING TUTORIALS Each chapter contains two Programming Tutorials that provide step-by-step instructions for using the chapter's concepts in an application. In most cases, the first tutorial in each chapter is easier than the second because it contains more detailed steps. Typically, one of the tutorial applications is a simple game, while the other is a business application. Game applications are used because research shows that the fun and exciting nature of games helps motivate students to learn.

PROGRAMMING EXAMPLE A Programming Example follows the Programming Tutorials in each chapter. The Programming Example shows the TOE chart and pseudocode used to plan the application. It also shows the user interface and Visual Basic code.

SUMMARY Each chapter contains a Summary section that recaps the concepts covered in the chapter.

KEY TERMS Following the Summary section in each chapter is a listing of the key terms introduced throughout the chapter, along with their definitions.

REVIEW QUESTIONS Each chapter contains Review Questions designed to test a student's understanding of the chapter's concepts.

PENCIL AND PAPER EXERCISES Following the Review Questions in each chapter are Pencil and Paper Exercises. The Exercises are designated as Modify This, Introductory, Intermediate, Advanced, Discovery, and Swat the Bugs. The Advanced and Discovery Exercises provide practice in applying cumulative programming knowledge. They also allow students to explore alternative solutions to programming tasks. The Swat the Bugs Exercises provide an opportunity for students to detect and correct errors in one or more lines of code.

COMPUTER EXERCISES The Computer Exercises, which follow the Pencil and Paper Exercises in each chapter, provide students with additional practice of the skills and concepts they learned in the chapter. The Exercises are designated as Modify This, Introductory, Intermediate, Advanced, Discovery, and Swat the Bugs. The Advanced and Discovery Exercises provide practice in applying cumulative programming knowledge. They also allow students to explore alternative solutions to programming tasks. The Swat the Bugs Exercises provide an opportunity for students to detect and correct errors in an existing application.

CASE PROJECTS At the end of each chapter are at least four Case Projects, one (or more) of which is a Think Tank Case Project. The Case Projects give the student the opportunity to independently synthesize and evaluate information, examine potential solutions, and make recommendations. Most of the Case Projects include a sample interface.

THINK TANK CASE PROJECTS Each chapter contains at least one Think Tank Case Project. These projects are more challenging than the other Case Projects.

New to this Edition!

NEW EXAMPLES, SAMPLE PROGRAMS, TUTORIALS, AND EXERCISES Each chapter has been updated with new examples, sample programs, tutorials, and/or exercises.

VIDEOS The videos that accompany the book have been updated from the previous edition.

VISUAL STUDIO COMMUNITY 2015 AND WINDOWS 10 The figures in the book and the steps in the Programming Tutorials assume you are using Visual Studio Community 2015 and a system running Microsoft Windows 10. Your screen and steps may be slightly different if you are using a different version of Visual Studio and/or Microsoft Windows.

STARTING AND CONFIGURING VISUAL STUDIO The How To boxes in Figures 1-1 and 1-2 in Chapter 1 contain the instructions for starting and configuring Visual Studio Community 2015 on a Windows 10 system. The instructions for starting Visual Studio Community on a Windows 8 system are contained in the VbReloaded2015\Chap01\Starting Using Windows 8.pdf file.

Instructor Resources

The following resources are available on the Instructor Companion Site (*sso.cengage.com*) to instructors who have adopted this book. Search for this title by ISBN, title, author, or keyword. From the Product Overview page, select the Instructor's Companion Site link to access your complementary resources.

INSTRUCTOR'S MANUAL The Instructor's Manual follows the text chapter by chapter to assist you in planning and organizing an effective, engaging course. The manual includes learning objectives, chapter overviews, ideas for classroom activities, and additional resources. A sample course **Syllabus** is also available.

TEST BANK Cengage Learning Testing Powered by Cognero is a flexible, online system that allows you to:

- author, edit, and manage test bank content from multiple Cengage Learning solutions
- create multiple test versions in an instant
- deliver tests from your LMS, your classroom or wherever you want

POWERPOINT PRESENTATIONS This book comes with Microsoft PowerPoint slides for each chapter. These are included as a teaching aid for classroom presentation, to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors are encouraged to customize the slides to fit their course needs, and may add slides to cover additional topics using the complete **Figure Files** from the text, also available on the Instructor Companion Site.

SOLUTION FILES Solutions to the Review Questions, Pencil and Paper Exercises, Computer Exercises, and Case Projects are available. The Solution Files also contain the sample programs that appear in the figures throughout the book.

DATA FILES Data Files are required to complete many of the computer activities in this book. They are available on the Instructor Companion Site as well as on CengageBrain.com.

Acknowledgments

Writing a book is a team effort rather than an individual one. I would like to take this opportunity to thank my team, especially Alyssa Pratt (Senior Content Developer), Heidi Aguiar (Project Manager), Jennifer Feltri-George (Senior Content Project Manager), Suzanne Huizenga (Proofreader), John Frietas (Quality Assurance), and the compositors at GEX. Thank you for your support, enthusiasm, patience, and hard work. Last, but certainly not least, I want to thank the following reviewers for their invaluable ideas and comments: Lorraine Bergkvist, University of Baltimore and Adam Lee, University of Maryland.

Diane Zak

Read This Before You Begin

Technical Information

Data Files

You will need data files to complete the computer activities in this book. Your instructor may provide the data files to you. You may obtain the files electronically at CengageBrain.com and then navigating to the page for this book.

Each chapter in this book has its own set of data files, which are stored in a separate folder within the VbReloaded2015 folder. The files for Chapter 1 are stored in the VbReloaded2015\Chap01 folder. Similarly, the files for Chapter 2 are stored in the VbReloaded2015\Chap02 folder. Throughout this book, you will be instructed to open files from or save files to these folders.

You can use a computer in your school lab or your own computer to complete the computer activities in this book.

Using Your Own Computer

To use your own computer to complete the computer activities in this book, you will need the following:

- A Pentium® 4 processor, 1.6 GHz or higher, personal computer running Microsoft Windows. This book was written using Microsoft Windows 10, and Quality Assurance tested using Microsoft Windows 8 and Microsoft Windows 10.
- Visual Studio Community 2015, Visual Studio Professional 2015, or Visual Studio Enterprise 2015 installed on your computer. This book was written and Quality Assurance tested using Visual Studio Community 2015. At the time of this writing, you can download a free copy of Visual Studio Community 2015 at *https://www.visualstudio.com/downloads/download-visual-studio-vs.*

To start and configure Visual Studio to match the figures and tutorial steps in this book:

1. Use the information in the How To box in Figure 1-1 (on page 10) to start Visual Studio Community 2015.
2. Use the information in the How To box in Figure 1-2 (on page 10) to configure Visual Studio Community 2015.

To install Microsoft Visual Basic PowerPacks 12.0:

1. Locate the vb_vbpowerpacks.exe file, which is contained in the VbReloaded2015\PowerPacks folder. Right-click the filename and then click Run as administrator. Click the Yes button.
2. Select the "I agree to the License Terms and Privacy Policy." check box. Either select or deselect the check box that asks if you want to join the Visual Studio Experience Improvement program. Click Install.
3. When the "Setup Successful!" message appears, click the Close button.
4. Start Visual Studio. Open the Toolbox window (if necessary) by clicking View on the menu bar and then clicking Toolbox. Right-click the Toolbox window and then click Add Tab. Type Visual Basic PowerPacks and press Enter.
5. Right-click the Visual Basic PowerPacks tab, and then click Choose Items. If necessary, click the .NET Framework Components tab in the Choose Toolbox Items dialog box.
6. In the Filter box, type PowerPacks. You may see one or more entries for the PrintForm control. Select all of Version 12's controls (PrintForm, DataRepeater, LineShape, OvalShape, and RectangleShape).
7. Click the OK button to close the Choose Toolbox Items dialog box. If the message "The following controls were successfully added to the toolbox but are not enabled in the active designer:" appears, click the OK button. The PowerPacks controls will not appear in the Toolbox window until you either create a new Visual Basic application or open an existing one. You will learn how to perform both of those tasks in Chapter 1.

Figures

The figures in this book reflect how your screen will look if you are using Visual Studio Community 2015 and a Microsoft Windows 10 system. Your screen may appear slightly different in some instances if you are using another version of either Visual Studio or Microsoft Windows.

Visit Our Web Site

Additional materials designed for this textbook might be available at CengageBrain.com. Search the site for more details.

To the Instructor

To complete the computer activities in this book, your students must use a set of data files. These files can be obtained on the Instructor Companion Site or at CengageBrain.com.

The material in this book was written using Visual Studio Community 2015 on a Microsoft Windows 10 system. It was Quality Assurance tested using Visual Studio Community 2015 on both a Windows 10 and a Windows 8 system.

OVERVIEW

An Introduction to Programming

After studying this Overview, you should be able to:

- Define the terminology used in programming
- Explain the tasks performed by a programmer
- Understand the employment opportunities for programmers and software developers
- Use the chapters effectively

Programming a Computer

In essence, the word **programming** means *giving a mechanism the directions to accomplish a task*. If you are like most people, you have already programmed several mechanisms, such as your digital video recorder (DVR), cell phone, or coffee maker. Like these devices, a computer is also a mechanism that can be programmed.

The directions (typically called instructions) given to a computer are called **computer programs** or, more simply, **programs**. The people who write programs are called **programmers**. Programmers use a variety of special languages, called **programming languages**, to communicate with the computer. Some popular programming languages are Visual Basic, C#, C++, and Java. In this book, you will use the Visual Basic programming language.

The Programmer's Job

When a company has a problem that requires a computer solution, typically it is a programmer who comes to the rescue. The programmer might be an employee of the company. Or, he or she might be a freelance programmer (or programming consultant), which is a programmer who works on temporary contracts rather than for a long-term employer.

Overview-Programmers

First the programmer meets with the user, who is the person (or people) responsible for describing the problem. In many cases, this person will also eventually use the solution. Depending on the complexity of the problem, multiple programmers may be involved, and they may need to meet with the user several times. Programming teams often contain subject matter experts, who may or may not be programmers. For example, an accountant might be part of a team working on a program that requires accounting expertise. The purpose of the initial meetings with the user is to determine the exact problem and to agree on the desired solution.

After the programmer and user agree on the solution, the programmer begins converting the solution into a computer program. During the conversion phase, the programmer meets periodically with the user to determine whether the program fulfills the user's needs and to refine any details of the solution. When the user is satisfied that the program does what he or she wants it to do, the programmer rigorously tests the program with sample data before releasing it to the user, who will test it further to verify that it correctly solves the problem. In many cases, the programmer also provides the user with a manual that explains how to use the program. As this process indicates, the creation of a good computer solution to a problem—in other words, the creation of a good program—requires a great deal of interaction between the programmer and the user.

Employment Opportunities

Overview-Programmer Qualities

When searching for a job in computer programming, you will encounter ads for "computer programmers" as well as for "computer software engineers." Although job titles and descriptions vary, computer software engineers typically are responsible for designing an appropriate solution to a user's problem, while computer programmers are responsible for translating the solution into a language that the computer can understand—a process called **coding**. Software engineering is a higher-level position that requires the ability to envision solutions. Using a construction analogy, software engineers are the architects, while programmers are the carpenters.

Keep in mind that depending on the employer as well as the size and complexity of the user's problem, the design and coding tasks may be performed by the same employee, no matter what his or her job title is. In other words, it's not unusual for a software engineer to code his or her solution or for a programmer to have designed the solution he or she is coding.

Programmers and software engineers need to have strong problem-solving and analytical skills, as well as the ability to communicate effectively with team members, end users, and other nontechnical personnel. Typically, software engineers are expected to have at least a bachelor's degree in software engineering, computer science, or mathematics, along with practical work experience, especially in the industry in which they are employed. Computer programmers usually need at least an associate's degree in computer science, mathematics, or information systems, as well as proficiency in one or more programming languages.

Computer programmers and software engineers are employed by companies in almost every industry, such as telecommunications companies, software publishers, financial institutions, insurance carriers, educational institutions, and government agencies. The Bureau of Labor Statistics predicts that employment of computer software engineers will increase by 22% from 2012 to 2022. The employment of computer programmers, on the other hand, will increase by 8% over the same period. In addition, consulting opportunities for freelance programmers and software engineers are expected to increase as companies look for ways to reduce their payroll expenses.

There is a great deal of competition for programming and software engineering jobs, so jobseekers will need to keep up to date with the latest programming languages and technologies. A competitive edge may be gained by obtaining vendor-specific or language-specific certifications, as well as knowledge of a prospective employer's business. More information about computer programmers and computer software engineers can be found on the U.S. Bureau of Labor Statistics Web site at *www.bls.gov.*

Using the Chapters Effectively

The chapters in this book teach you how to write programs using the Visual Basic programming language. Each chapter focuses on programming concepts, which are first introduced using simple examples and then utilized in larger applications at the end of the chapter.

Two Programming Tutorials follow the concepts section in each chapter. Each Programming Tutorial guides you through the process of creating an application using the concepts covered in the chapter. In most cases, the first tutorial in each chapter is easier than the second one because it contains more detailed steps. Some of the applications created in the Programming Tutorials are simple games, while others are business applications.

A Programming Example follows the Programming Tutorials in each chapter. The Programming Example contains a completed application that demonstrates the chapter concepts.

After reading the concepts section, be sure to complete one or both of the Programming Tutorials and the Programming Example. Doing this will help you complete the Computer Exercises and Case Projects at the end of the chapter. In addition, some of the Computer Exercises require you to modify the applications created in the Programming Tutorials and Programming Example.

Throughout each chapter, you will find How To boxes. Some How To boxes, like the one in Figure O-1, contain numbered steps that show you how to accomplish a task, such as how to open an existing solution in Visual Basic. You are not expected to follow the steps in these How To boxes while you are reading the chapter. Rather, these How To boxes are intended to provide a quick reference that you can use when completing the end-of-chapter Programming Tutorials, Programming Example, Exercises, and Case Projects. Feel free to skim these How To boxes and use them only if and when you need to do so. The same holds true for How To boxes containing bulleted items, like the one shown in Figure O-2.

HOW TO Open an Existing Solution

1. Click File on the menu bar and then click Open Project.
2. Locate and then open the application's solution folder. Click the solution filename, which ends with .sln. (The *sln* stands for *solution*.)
3. Click the Open button in the Open Project dialog box.
4. If the designer window is not open, click View on the menu bar and then click Designer. Or, you can right-click the form file's name in the Solution Explorer window and then click View Designer.

Note: To control the display of filename extensions, right-click the Start button on the Windows 10 taskbar, click Control Panel, click Appearance and Personalization, click File Explorer Options, and then click the View tab. Deselect the Hide extensions for known file types check box to show the extensions; or, select the check box to hide them.

Figure O-1 How to open an existing solution

HOW TO Start an Application

- Save the solution, click Debug on the menu bar, and then click Start Debugging.
- Save the solution and then press the F5 key on your keyboard.
- Save the solution and then click the Start button on the Standard toolbar.

Figure O-2 How to start an application

Other How To boxes contain information pertaining to a Visual Basic instruction. The How To box in Figure O-3, for example, contains the syntax and examples of assigning a value to an object's property while an application is running. You should study the information in these How To boxes while you are reading the chapter.

HOW TO Assign a Value to an Object's Property During Run Time

Syntax

object.property = expression

assignment operator

dot member access operator

Example 1

```
grossPayLabel.Width = 400
```

assigns the number 400 to the grossPayLabel's Width property

Example 2

```
companyLabel.Text = "Lakeside Hotels"
```

assigns the string "Lakeside Hotels" to the companyLabel's Text property

Example 3

```
goPictureBox.Visible = True
```

assigns the keyword `True` to the goPictureBox's Visible property

Figure 0-3 How to assign a value to an object's property during run time

The three most common learning styles are visual, auditory, and kinesthetic. Briefly, visual learners learn by watching, auditory learners learn by listening, and kinesthetic learners learn by doing. This book contains special elements designed specifically for each of the different learning styles. For example, each chapter contains videos for visual and auditory learners. The videos demonstrate and explain the concepts covered in the chapter. In addition to the Programming Tutorials, most chapters contain Try It! files for kinesthetic learners. The Try It! files allow the learner to practice with a concept before moving on to the next concept.

Overview Summary

- Programs are the step-by-step instructions that tell a computer how to perform a task.
- Programmers use various programming languages to communicate with the computer.
- In most cases, a programmer meets with the user several times to determine the exact problem to be solved and to agree on a solution. He or she also gets together periodically with the user to verify that the solution meets the user's needs and to refine any details.
- Programmers rigorously test a program with sample data before releasing the program to the user.
- It's not unusual for the same person to perform the duties of both a computer software engineer and a computer programmer.

Key Terms

Coding—the process of translating a solution into a language that the computer can understand

Computer programs—the directions given to computers; also called programs

Programmers—the people who write computer programs

Programming—the process of giving a mechanism the directions to accomplish a task

Programming languages—languages used to communicate with a computer

Programs—the directions given to computers; also called computer programs

CHAPTER 1

An Introduction to Visual Basic 2015

After studying Chapter 1, you should be able to:

1. Define some of the terms used in object-oriented programming
2. Create, start, and end a Visual Basic 2015 Windows Forms application
3. Manage the windows in the integrated development environment (IDE)
4. Set the properties of an object
5. Manipulate controls
6. Use label, button, and picture box controls
7. Use the options on the Format menu
8. Write Visual Basic code
9. Save, close, and open a solution
10. Run a project's executable file
11. Write an assignment statement
12. Print an application's code and interface
13. Find and correct a syntax error

Visual Basic 2015

In this book, you will learn how to create programs, called **applications**, using the Visual Basic 2015 programming language. Visual Basic 2015 is one of the languages built into Microsoft's newest integrated development environment: Visual Studio 2015. An **integrated development environment (IDE)** is an environment that contains all of the tools and features you need to create, run, and test your programs.

Visual Basic is an **object-oriented programming language**, which is a language that allows the programmer to use objects to accomplish a program's goal. An **object** is anything that can be seen, touched, or used. In other words, an object is nearly any *thing*. The objects in an object-oriented program can take on many different forms. Programs written for the Windows environment typically use objects such as check boxes, list boxes, and buttons. A payroll program, on the other hand, might utilize objects found in the real world, such as a time card object, an employee object, and a check object. Object-oriented programming is more simply referred to as **OOP**.

Every object in an object-oriented program is created from a **class**, which is a pattern that the computer uses to create the object. The class contains the instructions that tell the computer how the object should look and behave. An object created from a class is called an **instance** of the class and is said to be **instantiated** from the class. An analogy involving a cookie cutter and cookies is often used to describe a class and its objects: The class is the cookie cutter, and the objects instantiated from the class are the cookies. You will learn more about classes and objects throughout this book.

You can use Visual Basic to create applications for the Windows environment or for the Web. A Windows application has a Windows user interface and runs on a personal computer. A **user interface** is what the user sees and interacts with while an application is running. Examples of Windows applications include graphics programs, data-entry systems, and games. A Web application, on the other hand, has a Web user interface and runs on a server. You access a Web application using your computer's browser. Examples of Web applications include e-commerce applications available on the Internet and employee handbook applications accessible on a company's intranet. You can also use Visual Basic to create applications for tablet PCs and mobile devices, such as smartphones.

Starting and Configuring Visual Studio

Before you can create an application in Visual Basic, you must start Visual Studio. The steps for starting Visual Studio Community 2015 are listed in the How To box shown in Figure 1-1. Your steps may differ slightly if you are using a different edition of Visual Studio 2015. As mentioned in the Overview, you are not expected to follow the steps listed in a How To box right now. Rather, the steps are intended to be used as a quick reference while you are completing the Programming Tutorials, Programming Example, Exercises, and Case Projects located at the end of each chapter. For now, simply skim the How To box to see what it contains.

The Windows 10 Start button looks like this: ⊞.

HOW TO Start Visual Studio Community 2015

1. Click the Start button on the Windows 10 taskbar, click All apps, click the Visual Studio 2015 folder on the Start menu, and then click Visual Studio 2015.
2. *If the Choose Default Environment Settings dialog box appears*, click Visual Basic Development Settings and then click Start Visual Studio.

   If the Choose Default Environment Settings dialog box does not appear, you can select the appropriate settings (if necessary) as follows: click Tools on the menu bar, click Import and Export Settings, select the Reset all settings radio button, click the Next button, select the appropriate radio button, click the Next button, click Visual Basic, click the Finish button, and then click the Close button to close the Import and Export Settings Wizard dialog box.
3. To reset the IDE to the default layout for Visual Basic, click Window on the menu bar, click Reset Window Layout, and then click the Yes button.
4. To temporarily show/hide the underlined letters (called access keys) in the menu bar, press the Alt key on your keyboard. To permanently show/hide the access keys, right-click the Start button, click Control Panel, and then click Appearance and Personalization. In the Ease of Access Center section, click Turn on easy access keys. Select the Underline keyboard shortcuts and access keys check box, and then click the OK button. Close the Ease of Access Center window.

Note: The instructions for starting Visual Studio Community 2015 using Windows 8 are contained in the VbReloaded2015\Chap01\Starting Using Windows 8.pdf file.

Figure 1-1 How to start Visual Studio Community 2015

The How To box in Figure 1-2 contains the steps for configuring Visual Studio so that your screen and Programming Tutorial steps agree with the figures and tutorial steps in this book. You will be guided through performing these steps in Programming Tutorial 1 in this chapter. Figure 1-3 shows the Options dialog box mentioned in the steps.

HOW TO Configure Visual Studio Community 2015

1. Click Tools on the menu bar, and then click Options to open the Options dialog box. Click the Projects and Solutions node. Use the information shown in Figure 1-3 to select and deselect the appropriate check boxes.
2. Click the Debugging node and then deselect the Step over properties and operators (Managed only) check box. Also deselect the Enable Diagnostic Tools while debugging check box and the Show elapsed time PerfTip while debugging check box, which appear near the bottom of the list.
3. Click the OK button to close the Options dialog box.

Note: If you change your default environment settings *after* performing these three steps, you will need to perform the steps again.

Figure 1-2 How to configure Visual Studio Community 2015