8th
EDITION

PROGRAMMING WITH

MICROSOFT® VISUAL BASIC®

2017

DIANE ZAK

PROGRAMMING WITH MICROSOFT® VISUAL BASIC® 2017

EIGHTH EDITION

PROGRAMMING WITH MICROSOFT® VISUAL BASIC® 2017

DIANE ZAK

CENGAGE

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

CENGAGE

***Programming with Microsoft® Visual Basic® 2017*, Eighth Edition**
Diane Zak

SVP, GM Science, Technology & Math: Balraj S. Kalsi

Product Director: Kathleen McMahon

Product Team Manager: Kristin McNary

Associate Content Developer: Brianna Vorce

Associate Product Manager: Kate Mason

Senior Director, Development: Julia Caballero

Product Assistant: Jake Toth

Marketing Director: Michele McTighe

Marketing Manager: Stephanie Albracht

Senior Production Director: Wendy Troeger

Production Director: Patty Stephan

Senior Content Project Manager: Jennifer K. Feltri-George

Art Director: Diana Graham

Cover image(s): saemilee/DigitalVision Vectors/ Getty Images

Cover Designer: Lisa Buckley

Unless otherwise noted all screenshots are courtesy of Microsoft Corporation

Open Clip art source: OpenClipArt

Printed in the United States of America
Print Number: 01 Print Year: 2017

Library of Congress Control Number: 2017940681

Softcover ISBN-13: 978-1-337-10212-4
LLF ISBN: 978-1-337-68573-3

Cengage
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.

To learn more about Cengage Learning, visit **www.cengage.com**

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

Brief Contents

Contents

Preface

Programming with Microsoft Visual Basic 2017, Eighth Edition uses Visual Basic 2017, an object-oriented language, to teach programming concepts. This book is designed for a beginning programming course. However, it assumes that students are familiar with basic Windows skills and file management.

Organization and Coverage

Programming with Microsoft Visual Basic 2017, Eighth Edition contains 13 chapters that present concepts along with hands-on instruction; it also contains 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. GUI design skills, OOP concepts, and planning tools (such as Planning Charts, pseudocode, and flowcharts) are covered in the book. The chapters show students 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 also learn how to create and manipulate variables, constants, strings, sequential access files, classes, and arrays. In Chapters 11 and 12, students learn how to create SQL Server databases and then use them in applications. They also learn how to use Structured Query Language (SQL) to access specific fields and records from the database and also to create calculated fields. Chapter 13 shows students how to create both static and dynamic Web Site applications.

Appendix A contains a summary of the GUI design guidelines mentioned in the chapters. Appendix B contains additional topics that can be covered along with (or anytime after) a specified chapter. Appendix C teaches students how to locate and correct errors in their code. The appendix shows students how to step through their code and also how to create breakpoints. The Visual Basic 2017 Cheat Sheet contained in Appendix D summarizes important concepts covered in the chapters, such as the syntax of statements, methods, and so on. The Cheat Sheet provides a convenient place for students to locate the information they need as they are creating and coding their applications. Appendix E contains Case Projects that can be assigned after completing specific chapters in the book.

Approach

Each chapter in *Programming with Microsoft Visual Basic 2017, Eighth Edition* contains two lessons titled Focus on the Concepts and Apply the Concepts. Each lesson has its own set of objectives. The Focus lessons concentrate on programming concepts, using examples along with sample applications designed to reinforce the concepts being taught. The Apply the Concepts lessons show students how to apply the concepts from the chapter's Focus lesson in different ways. The Apply lessons also expand on the concepts taught in the Focus lesson. Both lessons provide tutorial-style steps that guide the student on coding, running, and testing applications. Each sample application allows the student to observe how the current concept can be used before the next concept is introduced.

Features

Programming with Microsoft Visual Basic 2017, Eighth 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.

YOU DO IT! BOXES These boxes provide simple applications that allow students to demonstrate their understanding of a concept before moving on to the next concept.

OPTION STATEMENTS All applications include the Option Explicit, Option Strict, and Option Infer statements.

START HERE ARROWS These arrows indicate the beginning of a tutorial steps section in the book.

FIGURES Figures that introduce new statements, functions, or methods contain both the syntax and examples of using the syntax. Including the syntax in the figures makes the examples more meaningful, and vice versa.

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.

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 list 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.

New to This Edition!

NEW CHAPTERS All of the chapters in the book have been revamped. A list of the changes made to the chapters is available via the optional MindTap for this text.

LESSONS Each chapter is divided into two lessons: Focus on the Concepts and Apply the Concepts. Each lesson has its own set of objectives. The Focus lessons introduce programming concepts, which are illustrated with code examples and sample applications. The Apply lessons show students how to apply the concepts from the chapter's Focus lesson in different ways. The Apply lessons also expand on the concepts taught in the Focus lesson. Both lessons provide tutorial-style steps that guide the student on coding, running, and testing applications. The applications allow students to observe how the current concept can be used before the next concept is introduced.

MINI-QUIZ BOXES Mini-quiz boxes are strategically placed to test students' knowledge at various points in the chapter. Answers to the quiz questions are printed upside down in the boxes, allowing students to determine whether they have mastered the material covered thus far before continuing with the chapter.

SQL SERVER DATABASES AND SQL The book includes two chapters (Chapters 11 and 12) on SQL Server databases and SQL.

NEW EXERCISES The Review Questions in each chapter are followed by Exercises, which provide students with additional practice of the skills and concepts they learned in the chapter. The Exercises are designated as INTRODUCTORY, INTERMEDIATE, ADVANCED, ON YOUR OWN, and FIX IT. The ON YOUR OWN Exercises encourage students to challenge and independently develop their own programming skills while exploring the capabilities of Visual Basic 2017. Students are given minimum guidelines to follow when creating the ON YOUR OWN applications. The FIX IT Exercises provide an opportunity for students to detect and correct errors in an application's code.

APPENDICES Appendix A summarizes the GUI design guidelines mentioned in the chapters, making it easier for the student to follow the guidelines when designing an application's interface. Appendix B contains additional topics that can be covered along with (or anytime after) a specified chapter. Appendix C teaches students how to locate and correct errors (syntax, logic, and run time) in their code. The appendix shows students how to step through their code and also how to create breakpoints. Appendix D contains a Cheat Sheet that summarizes important concepts covered in the chapters, such as the syntax of statements, methods, and so on. The Cheat Sheet provides a convenient place for students to locate the information they need as they are creating and coding their applications. Appendix E contains Case Projects that can be assigned after completing specific chapters in the book.

UPDATED VIDEOS These notes direct students to videos that accompany many chapters in the book. The videos explain and/or demonstrate one or more of the chapter's concepts. The videos have been revised from the previous edition and are available via the optional MindTap for this text.

Steps and Figures

The tutorial-style steps and figures in the book assume you are using Microsoft Visual Studio Community 2017 and a system running Microsoft Windows 10. Your screen may appear slightly different in some instances if you are using a different version of Microsoft Windows.

Instructor Resources

The following teaching tools are available for download at our Instructor Companion Site. Simply search for this text at sso.cengage.com. An instructor login is required.

INSTRUCTOR'S MANUAL The Instructor's Manual that accompanies this textbook includes additional instructional material to assist in class preparation, including items such as Sample Syllabi, Chapter Outlines, Technical Notes, Lecture Notes, Quick Quizzes, Teaching Tips, Discussion Topics, and Additional Case Projects.

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 offers 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 can add their own slides for additional topics they introduce to the class.

SOLUTION FILES Solutions to the chapter applications and the end-of-chapter Review Questions and Exercises are provided.

DATA FILES Data Files are necessary for completing the computer activities in this book. Data Files can also be downloaded by students at CengageBrain.com.

MindTap

MindTap is a personalized teaching experience with relevant assignments that guide students to analyze, apply, and improve thinking, allowing you to measure skills and outcomes with ease.

- Personalized teaching: Becomes yours with a Learning Path that is built with key student objectives. Control what students see and when they see it. Use it as-is or match to your syllabus exactly—hide, rearrange, add, and create your own content.
- Guide students: A unique learning path of relevant readings, multimedia and activities that move students up the learning taxonomy from basic knowledge and comprehension to analysis and application.
- Promote better outcomes: Empower instructors and motivate students with analytics and reports that provide a snapshot of class progress, time in course, engagement and completion rates.

The MindTap for *Programming with Microsoft Visual Basic 2017* includes videos, study tools, and interactive quizzing, all integrated into a full eReader that contains the full content from the printed text.

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 Jennifer Feltri-George (Senior Content Project Manager), Prathiba Rajagopal (Senior Project Manager), Nicole Spoto (Quality Assurance), Alyssa Pratt (Senior Content Developer), Brianna Vorce (Associate Content Developer), and the compositors at SPi Global. Thank you for your support, enthusiasm, patience, and hard work. And a special thank you to all of the past and present reviewers, as well as to Sally Douglas (College of Central Florida), who suggested the YOU DO IT! boxes several editions ago.

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 VB2017 folder. The files for Chapter 1 are stored in the VB2017\Chap01 folder. Similarly, the files for Chapter 2 are stored in the VB2017\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 steps and Exercises 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 and Quality Assurance tested using Microsoft Windows 10.
- Microsoft Visual Studio 2017 installed on your computer. (You need to install the .NET desktop development component to complete Chapters 1 through 12. You need to install the ASP.NET and web development component to complete Chapter 13.) This book was written and Quality Assurance tested using Microsoft Visual Studio Community 2017. At the time of this writing, you can download a free copy of the Community Edition at *https://www.visualstudio.com/downloads*.

To control the display of filename extensions in Windows 10:

1. Press and hold down the Windows logo key on your keyboard as you tap the letter x. (Or, right-click the Windows Start button on the taskbar.) Click Control Panel, click Appearance and Personalization, click File Explorer Options, and then click the View tab.
2. Deselect the Hide extensions for known file types check box to show the extensions; or, select the check box to hide them. Click the OK button and then close the Appearance and Personalization window.

To always display the underlined letters (called access keys) in Windows 10:

1. Press and hold down the Windows logo key on your keyboard as you tap the letter x. (Or, right-click the Windows Start button on the taskbar.) Click Control Panel and then click Appearance and Personalization.
2. In the Ease of Access Center section, click Turn on easy access keys, and then select the Underline keyboard shortcuts and access keys check box. Click the OK button and then close the Ease of Access Center window.

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

1. Use the steps on Pages 9 through 11 in Chapter 1.
2. If you are using the Professional or Enterprise editions of Visual Studio, you may also need to click Tools, click Options, expand the Text Editor node, expand the All Languages node, click CodeLens, and then deselect Enable CodeLens.

Figures

The figures in this book reflect how your screen will look if you are using Microsoft Visual Studio Community 2017 and a Microsoft Windows 10 system. Your screen may appear slightly different in some instances if you are using another version of either Microsoft Visual Studio or Microsoft Windows.

Visit Our Web Site

Additional materials designed for this textbook might be available at CengageBrain.com. Search this 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 and Quality Assurance tested using Microsoft Visual Studio Community 2017 on a Microsoft Windows 10 system.

CHAPTER 1

An Introduction to Visual Studio 2017 and Visual Basic

In this chapter's Focus on the Concepts lesson, you will learn the definitions for many of the terms used by programmers. The lesson also introduces you to Microsoft's newest integrated development environment (IDE): Visual Studio 2017. The IDE contains the latest version of the Visual Basic programming language. You will use the IDE and language to create the applications in this book. As stated in the Read This Before You Begin section of this book, the steps and figures in this book assume you are using the Community edition of Visual Studio 2017. Your steps and screen might differ slightly in some instances if you are using a different edition of Visual Studio 2017.

In the Apply the Concepts lesson, you will apply the concepts covered in the Focus lesson. The Apply lesson is designed to help you get comfortable with both the Visual Studio IDE and the Visual Basic programming language.

FOCUS ON THE CONCEPTS LESSON

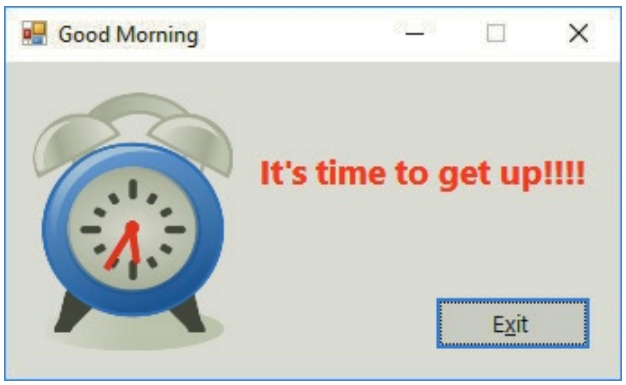
Concepts covered in this lesson:

- F-1 Computer programming terminology
- F-2 The programmer's job
- F-3 The Visual Basic programming language
- F-4 The Visual Studio IDE
- F-5 Assigning names to objects

F-1 Computer Programming Terminology

In essence, the word **programming** means *giving a mechanism the directions to accomplish a task*. When the mechanism is a computer, the directions are typically referred to as instructions. A set of instructions that tells a computer how to accomplish a task is called a **computer program** or, more simply, a **program**.

Programs are written by **programmers** using a variety of special languages called **programming languages**. Some popular programming languages are Visual Basic, C#, C++, and Java. In this book, you will write your programs using the Visual Basic programming language, which is built into Microsoft's newest integrated development environment: Visual Studio 2017. 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. You also will use the IDE to create graphical user interfaces for your programs. A **graphical user interface** (or **GUI**) is what the person using your program (referred to as the user) sees and interacts with while your program is running. The user interface and its program instructions are referred to as an **application**. Figure 1-1 shows the user interface and program instructions for the Good Morning application.




```
1   Public Class frmMain
2       Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles btnExit.Click
3           Me.Close()
4       End Sub
5
6       Private Sub tmrGetUp_Tick(sender As Object, e As EventArgs) Handles tmrGetUp.Tick
7           ' Blink the message.
8
9           lblMessage.Visible = Not lblMessage.Visible
10      End Sub
```

Figure 1-1 Good Morning application

START HERE

To run the Good Morning application:

1. Use Windows to locate and then open the VB2017\Chap01 folder on your computer's hard disk or on the device designated by your instructor. Double-click **Good Morning.exe** in the list of filenames. (Depending on how Windows is set up on your computer, you might not see the .exe extension on the filename. Refer to the Read This Before You Begin section to learn how to show filename extensions.) The application's user interface appears on the screen with a blinking "It's time to get up!!!!" message.
2. Click the **Exit** button to close the application.

F-2 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, who works on temporary contracts rather than for a long-term employer.

First, the programmer meets with the person (or people) responsible for describing the problem. This person might be the one who will eventually use the solution; or he or she might be a software developer, who serves as an intermediary between the user and the programmer. The software developer will meet with the user and then outline the problem specification for the programmer. After the programmer understands the problem, he or she will begin planning an appropriate solution. After the planning is complete, the programmer will translate the solution into computer instructions—a process called **coding**. The programmer then will test the program rigorously with sample data to make sure it works both correctly and to the user's satisfaction. Depending on the complexity of the problem, multiple programmers might be involved in the planning and coding phases. Programming teams often contain subject matter experts, who might or might not be programmers. For example, an accountant might be part of a team working on a program that requires accounting expertise.

F-3 The Visual Basic Programming Language

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. In object-oriented programming, or **OOP**, an **object** is anything that can be seen, touched, or used. In other words, an object is nearly any *thing*. Programs written for the Windows environment typically use objects such as check boxes, list boxes, and buttons.

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.

In the following set of steps, you will run two Visual Basic applications that employ many of the objects you will learn about in this book.

START HERE

To run the applications:

1. If necessary, open the VB2017\Chap01 folder. Double-click **Payment.exe**. After a few moments, the Monthly Payment Calculator application shown in Figure 1-2 appears on the screen. (If some of the letters on your screen are not underlined, press the Alt key.) The interface contains a text box, a list box, buttons, radio buttons, and labels. You can use the application to calculate the monthly payment for a car loan.

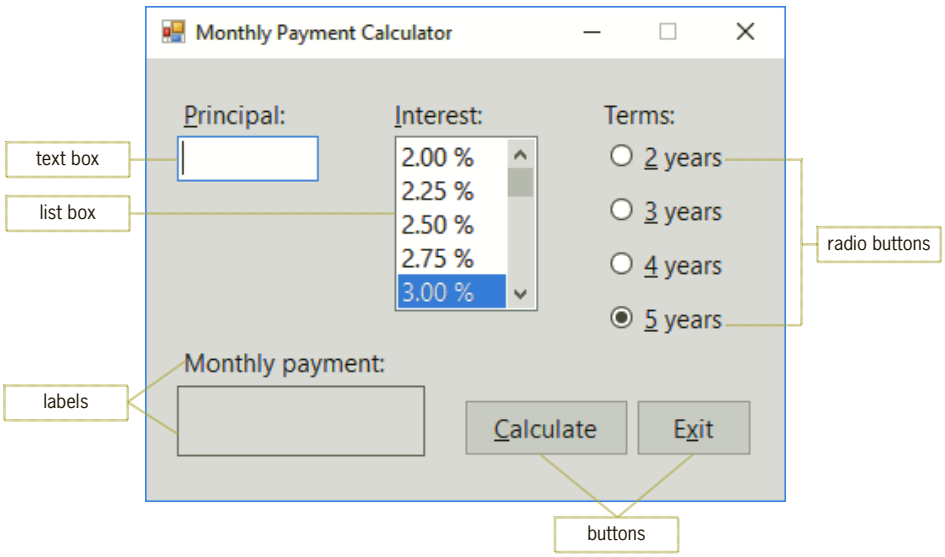


Figure 1-2 Monthly Payment Calculator application

2. First, you will use the application to calculate the monthly payment for a $15,000 loan at 3.5% interest for five years. Type **15000** in the Principal text box. Scroll down the Interest list box and then click **3.50 %**. Finally, click the **Calculate** button. The application indicates that your monthly payment would be $272.88. See Figure 1-3.

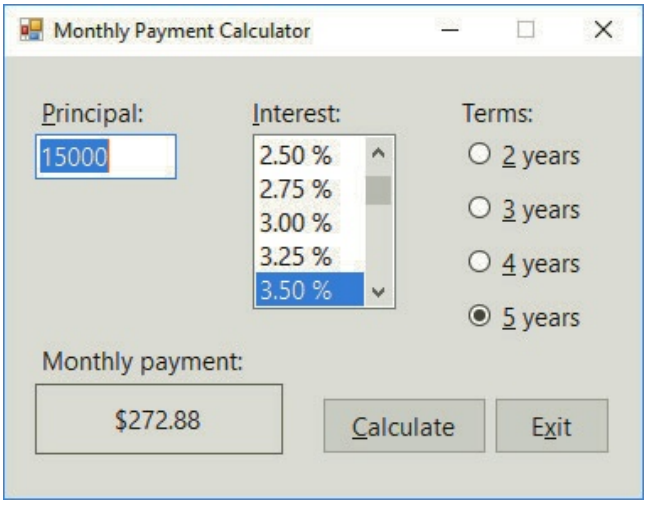


Figure 1-3 Computed monthly payment

3. Next, you will determine what your monthly payment would be if you borrowed $4,500 at 2.75% interest for four years. Type **4500** in the Principal text box, click **2.75 %** in the Interest list box, click the **4 years** radio button, and then click the **Calculate** button. The Monthly payment box shows $99.11.

4. Click the **Exit** button to close the application.
5. Now double-click **Einstein.exe**. (The file is located in the VB2017\Chap01 folder.) Click the **Show equation** button to display Einstein's famous equation. See Figure 1-4. The interface contains picture boxes and buttons. (If some of the letters on your screen are not underlined, press the Alt key.)

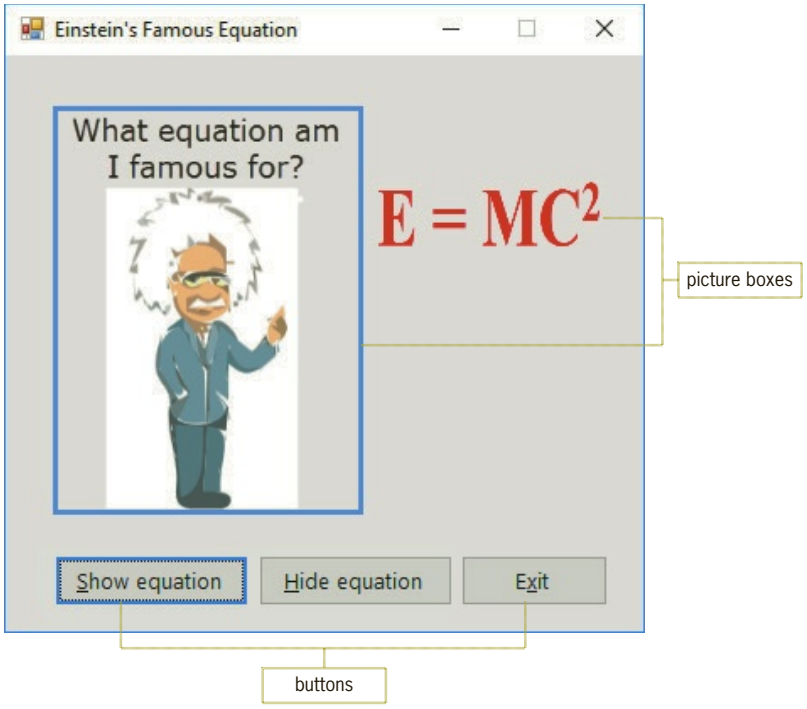


Figure 1-4 Equation shown in the interface

6. Click the **Hide equation** button to hide the equation, and then click the **Exit** button to close the application.

Mini-Quiz 1-1

1. What is a computer program?
2. What is a GUI?
3. What is the process of translating a solution into a computer program called?
4. In object-oriented programming, what is a class?

1) A set of instructions that tells a computer how to accomplish a task. **2)** Stands for Graphical User Interface. It is what the user sees and interacts with when using an application. **3)** Coding **4)** A pattern used to instantiate (create) an object.

F-4 The Visual Studio IDE

The Visual Studio IDE contains many different windows, each with its own special purpose. The four windows you will use most often when designing your user interfaces are shown in Figure 1-5.

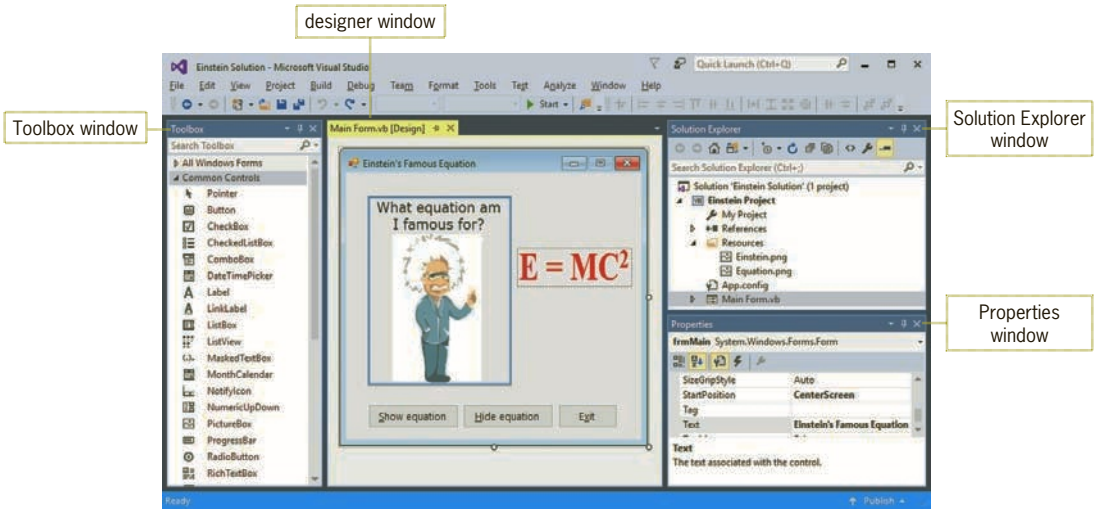


Figure 1-5 Visual Studio IDE

The **designer window** is where you create (or design) your application's GUI. A Windows Form object, or form, appears in the designer window shown in Figure 1-5. A **form** is the foundation for the user interface in an application created for the Windows environment. As you learned earlier, all objects in an object-oriented program are instantiated (created) from a class. A form, for example, is an instance of the Windows Form class. The form (an object) is automatically instantiated for you when you create a Windows Forms application in Visual Basic.

You use the **Toolbox window** to add other objects, called **controls**, to the form. Each tool listed in the Toolbox window represents a class. You add an object by clicking its corresponding tool (class) in the toolbox and then dragging it with your mouse pointer to the form. When you drag the tool to the form, Visual Basic creates (instantiates) an instance of the class (an object) and places it on the form. For example, the two picture box objects shown in Figure 1-5 were instantiated (created) by dragging the PictureBox tool from the toolbox to the form. Similarly, the three button objects were instantiated using the Button tool.

Each object has a set of attributes that determine its appearance and behavior. The attributes, called **properties**, are listed in the **Properties window** when the object is selected in the designer window. In Figure 1-5, the form is selected, and the names of its properties (such as StartPosition and Text), along with their values (CenterScreen and Einstein's Famous Equation), appear in the Properties window. You can use the Properties window to change the value of an object's property. For example, you can use it to change the form's Text property, which appears in the form's title bar, from Einstein's Famous Equation to Guess My Equation.

Windows applications in Visual Basic are composed of solutions, projects, and files. A solution is a container that stores the projects and files for an entire application. A project is also a container, but it stores only the files associated with that particular project. The **Solution Explorer** window displays a list of the projects contained in the current solution and the items contained in each project. The Solution Explorer window shown in Figure 1-5 indicates that the Einstein Solution contains the Einstein Project, which contains several items. The Einstein.png

and Equation.png items are the names of files on your disk. These files contain the images that appear in the picture boxes on the form. The Main Form.vb item is also the name of a file on your disk. The Main Form.vb file stores the program instructions (**code**) that tell the three buttons how to respond when the user clicks them. You enter the code in the **Code Editor window**, which is shown in Figure 1-6.

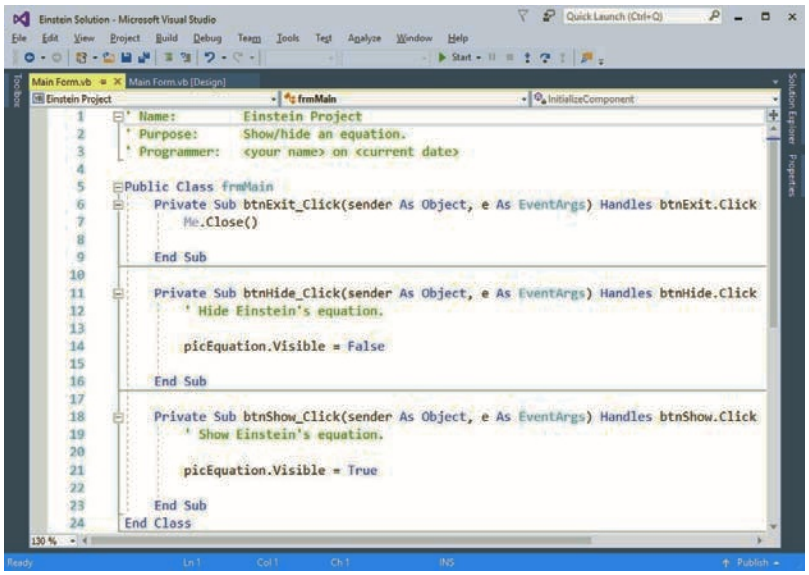


Figure 1-6 Code Editor window

At this point, you are not expected to understand the contents of the Code Editor window in Figure 1-6; you will learn about the contents in this chapter's Apply the Concepts lesson. However, briefly, the green lines of text are comments and are not executed by the computer when the application is run; they serve simply to internally document the program. The code on Lines 6 through 9 tell the computer to close (end) the application when the Exit button is clicked. The code on Lines 11 through 16 indicate that the computer should hide the equation picture box when the Hide equation button is clicked. The code on Lines 18 through 23 tell the computer to show the equation picture box when the Show equation button is clicked.

F-5 Assigning Names to Objects

As mentioned earlier, each object has a set of properties attached to it. One of the most important of these properties is the Name property. This is because you use the **Name property** to refer to the object in code. The code in Figure 1-6, for example, refers to objects named frmMain, btnExit, btnHide, btnShow, and picEquation. Figure 1-7 lists the rules and conventions this book will follow when naming objects. Typically, you assign names to only objects that are either coded or referred to in code.

Camel case refers to the fact that the uppercase letters appear as "humps" in the name because they are taller than the lowercase letters.

Naming rules (these are required by Visual Basic)

1. Each object must have a unique name.
2. Each name must begin with a letter and contain only letters, numbers, and the underscore character.

Naming conventions used in this book

1. Each name will begin with an ID of three (or more) characters that represents the object's type—for example, *frm* for a form, *btn* for a button, and *txt* for a text box.
2. The remaining characters after the ID will indicate the object's purpose.
3. Each name will be entered using camel case: the ID will be in lowercase, and the first letter in each subsequent word in the name will be capitalized.

Examples

frmMain	the main form in a project
btnExit	a button that ends the application when clicked
txtFirstName	a text box for entering a customer's first name
picEquation	a picture box that displays an image of an equation
lblTotalDue	a label that displays the total amount a customer owes
chkDiscount	a check box for specifying whether a customer gets a discount

Figure 1-7 Rules and conventions for naming objects

Mini-Quiz 1-2

1. Which window in the IDE lists the tools you can use to add objects to a form?
2. While designing an interface, which window in the IDE allows you to change the default value of an object's property?
3. Using the naming rules and conventions listed in Figure 1-7, which of the following are valid names and which are not? Explain why the names are not valid.
   a. lblTotal
   b. txtFirst.Name
   c. lblCity&State
   d. btnCalc Total
   e. txtFirstQuarter

1) Toolbox **2)** Properties **3)** Answers a and e are valid. Answer b is invalid because it contains a period. Answer c is invalid because it contains an ampersand. Answer d is invalid because it contains a space.

APPLY THE CONCEPTS LESSON

After studying this lesson, you should be able to:

- A-1 Start and configure Visual Studio Community 2017
- A-2 Create a Windows Forms application
- A-3 Manage the windows in the IDE
- A-4 Change a form file's name
- A-5 Change the properties of a form
- A-6 Save a solution
- A-7 Close and open a solution
- A-8 Add a control to a form
- A-9 Use the Format menu
- A-10 Lock the controls on the form
- A-11 Start and end an application
- A-12 Enter code and comments in the Code Editor window
- A-13 Print an application's code and interface
- A-14 Exit Visual Studio and run an executable file

Videos:

- Ch01-Adding a Control
- Ch01-Format Menu
- Ch01-Snipping Tool

A-1 Start and Configure Visual Studio Community 2017

In this Apply lesson, you will create the Einstein's Famous Equation application that you viewed in the Focus lesson. First, you need to start and configure Visual Studio Community 2017, which contains the Visual Basic language. (Keep in mind that your steps might differ slightly if you are using a different edition of Visual Studio 2017.)

To start Visual Studio Community 2017:

START HERE

1. Click the **Start** button on the Windows 10 taskbar. Locate and then click **Visual Studio 2017** in the Start menu.
2. Click **Tools** on the menu bar, click **Import and Export Settings**, select the **Reset all settings** radio button, click the **Next** button, select the **No, just reset settings, overwriting my current settings** radio button, click the **Next** button, click **Visual Basic**, and then click the **Finish** button. Click the **Close** button to close the Import and Export Settings Wizard dialog box.
3. Click **Window** on the menu bar, click **Reset Window Layout**, and then click the **Yes** button. When you start Visual Studio 2017, your screen will appear similar to Figure 1-8. However, your menu bar might not contain the underlined letters, called access keys. You can show/hide the access keys by pressing the Alt key on your keyboard.