Visual Basic 2019

Made Easy

By Dr.Liew

Disclaimer

Visual Basic 2019 Made Easy is an independent publication and is not affiliated with, nor has it been authorized, sponsored, or otherwise approved by Microsoft Corporation.

Trademarks

Microsoft, Visual Basic, Excel and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks belong to their respective owners.

Liability

The purpose of this book is to provide basic guides for people interested in Visual Basic 2019 programming. Although every effort and care has been taken to make The information as accurate as possible, the author shall not be liable for any error, Harm or damage arising from using the instructions given in this book.

Acknowledgement

I would like to express my sincere gratitude to the many people who have made their contributions in one way or another to the successful publication of this book.

My special thanks go to my children Xiang, Yi and Xun who have contributed their ideas and help in editing this book. I would also like to appreciate the support provided by my beloved wife Kim Huang and my youngest daughter Yuan. I would also like to thank the millions of readers who have visited my **Visual Basic Tutorial** website at **vbtutor.net** for their support and encouragement.

About the Author

Dr. Liew Voon Kiong holds a bachelor's degree in Mathematics, a master's degree in Management and a doctorate in Business Administration. He has been involved in Visual Basic programming for more than 20 years. He created the popular online Visual Basic Tutorial at **www.vbtutor.net**, which has attracted millions of visitors since 1996. It has consistently been one of the highest ranked Visual Basic websites.

To provide more support for Visual Basic students, teachers, and hobbyists, Dr. Liew has written this book to complement the free Visual Basic 2019 tutorial with much more content. He is also the author of the Visual Basic Made Easy series, which includes **Visual Basic 6 Made Easy, Visual Basic 2008 Made Easy, Visual Basic 2010 Made Easy, Visual Basic 2013 Made Easy, Visual Basic 2015 Made Easy, Visual Basic 2017 Made Easy** and **Excel VBA Made Easy.** Dr. Liew's books have been used in high school and university computer science courses all over the world.

Table of Contents

Chapter 1 Introduction

1.1 A Brief History of Visual Basic

Visual Basic is a third-generation event-driven programming language first released by Microsoft in 1991. The final version of the classic Visual Basic was Visual Basic 6. Visual Basic 6 is a user-friendly programming language designed for beginners. It enables anyone to develop GUI Windows applications easily. Many developers still favor VB6 over its successor VB.NET.

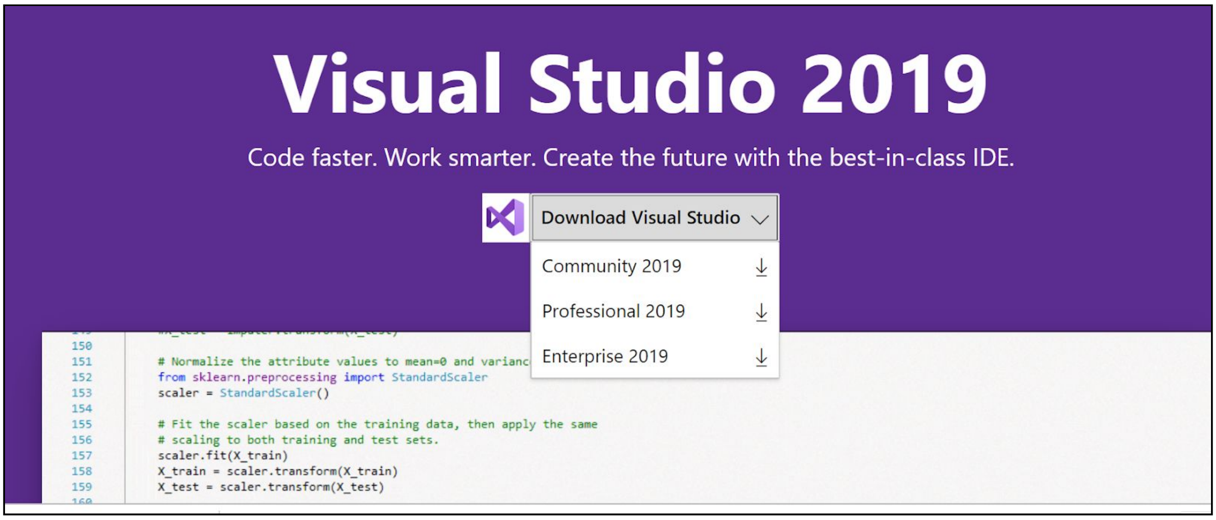
In 2002, Microsoft released Visual Basic.NET (VB.NET) to replace Visual Basic 6. Thereafter, Microsoft declared VB6 a legacy programming language in 2008. However, Microsoft still provides some form of support for VB6. VB.NET is a fully object-oriented programming language implemented in the .NET Framework. It was created to cater for the development of the web as well as mobile applications. Subsequently, Microsoft has released many versions of VB.NET. They are Visual Basic 2005, Visual Basic 2008, Visual Basic 2010, Visual Basic 2012, Visual Basic 2013, Visual Basic 2015, Visual Basic 2017 and Visual Basic 2019. Although the .NET portion was discarded in 2005, all versions of the Visual Basic programming language released since 2002 are regarded as the VB.NET programming language

Microsoft has released Visual Studio 2019 in early 2019. VS 2019 allows you to code in different programming languages and different platforms, Visual Basic 2019 is one of them. The other Programming languages are C# C++, F#, JavaScript, Java and Python. Visual Basic 2019 is the latest version of the VB.NET programming language released by Microsoft.

1.2 Installation of Visual Studio 2019

You can download the free version of Visual Studio 2019 from the following link: https://visualstudio.microsoft.com/vs/

Clicking the link brings up the Visual Studio 2019 download page, as shown in Figure 1.1

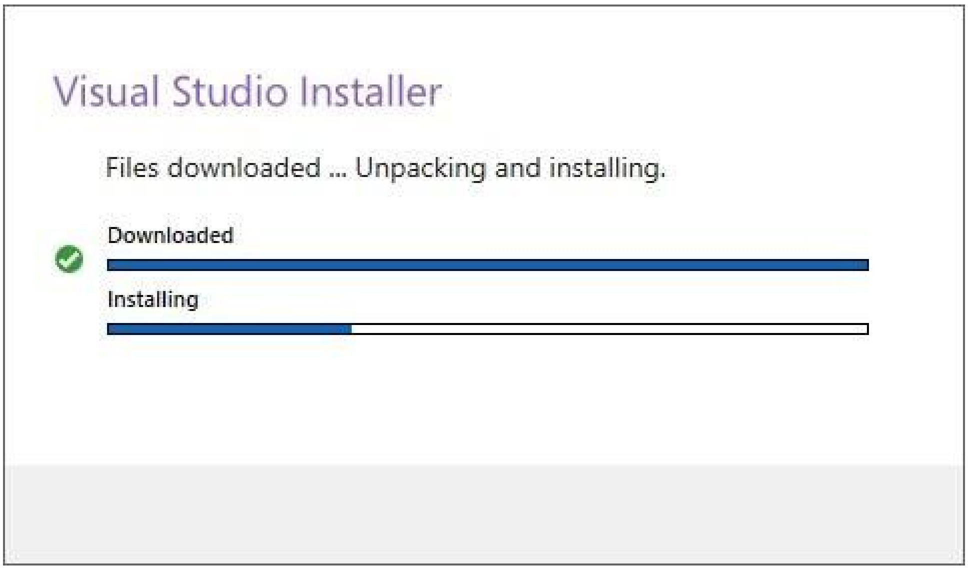


```
# Normalize the attribute values to mean=0 and varianc
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

# Fit the scaler based on the training data, then apply the same
# scaling to both training and test sets.
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Figure 1.1

You can choose the free Visual Studio Community 2019 or the Full-featured Professional 2019 and End-to-End solution Enterprise 2019 to download. The free version provides full-featured IDE for students, open source community and individuals. Since this book was written based on the free version, proceed to download the free Visual Studio 2019 Community, select community and download the installer file. The downloaded installer file will appear on your Windows 10 taskbar. Click it to install Visual Studio 2019. Clicking the Visual Studio 2019 Installer will start downloading, unpacking and installing the files necessary for the installation of Visual Studio 2019, as shown in Figure 1.2



Visual Studio Installer

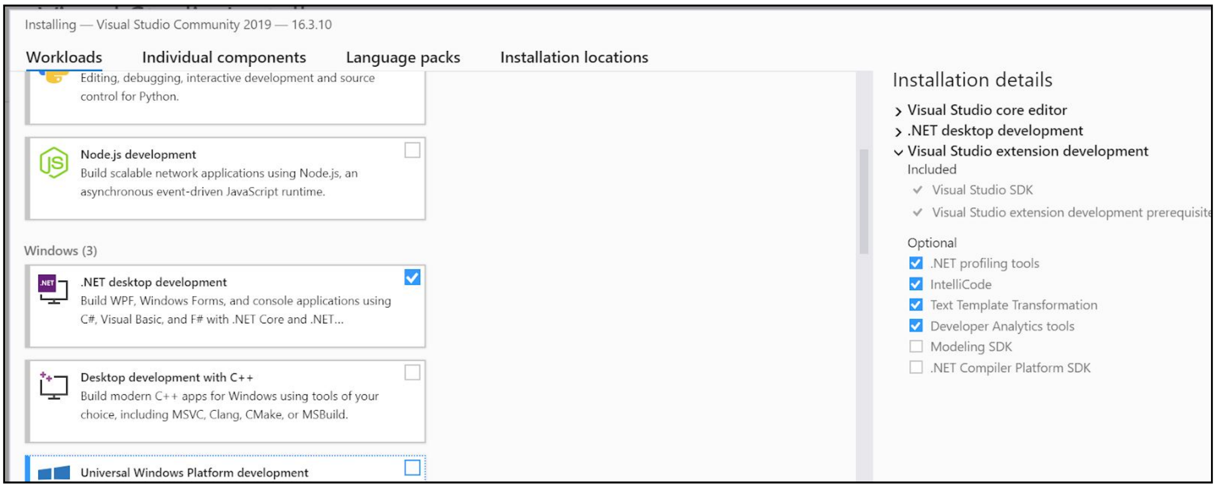
Files downloaded ... Unpacking and installing.

✓ Downloaded

Installing

Figure 1.2

You will see several status screens that show the progress of the installation. After the installer has finished installing, it is time to pick the feature set that you wish to install, as shown in Figure 1.3. Since we are focusing on developing Visual Basic 2019 desktop app, we will select the .NET desktop development component. After making your selections, click install.



Installing — Visual Studio Community 2019 — 16.3.10

Editing, debugging, interactive development and source control for Python.

Node.js development □
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Windows (3)

.NET desktop development ☑
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET...

Desktop development with C++ □
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

Universal Windows Platform development □

Installation details

- › Visual Studio core editor
- › .NET desktop development
- ⌄ Visual Studio extension development

Included

- ✓ Visual Studio SDK
- ✓ Visual Studio extension development prerequisite

Optional

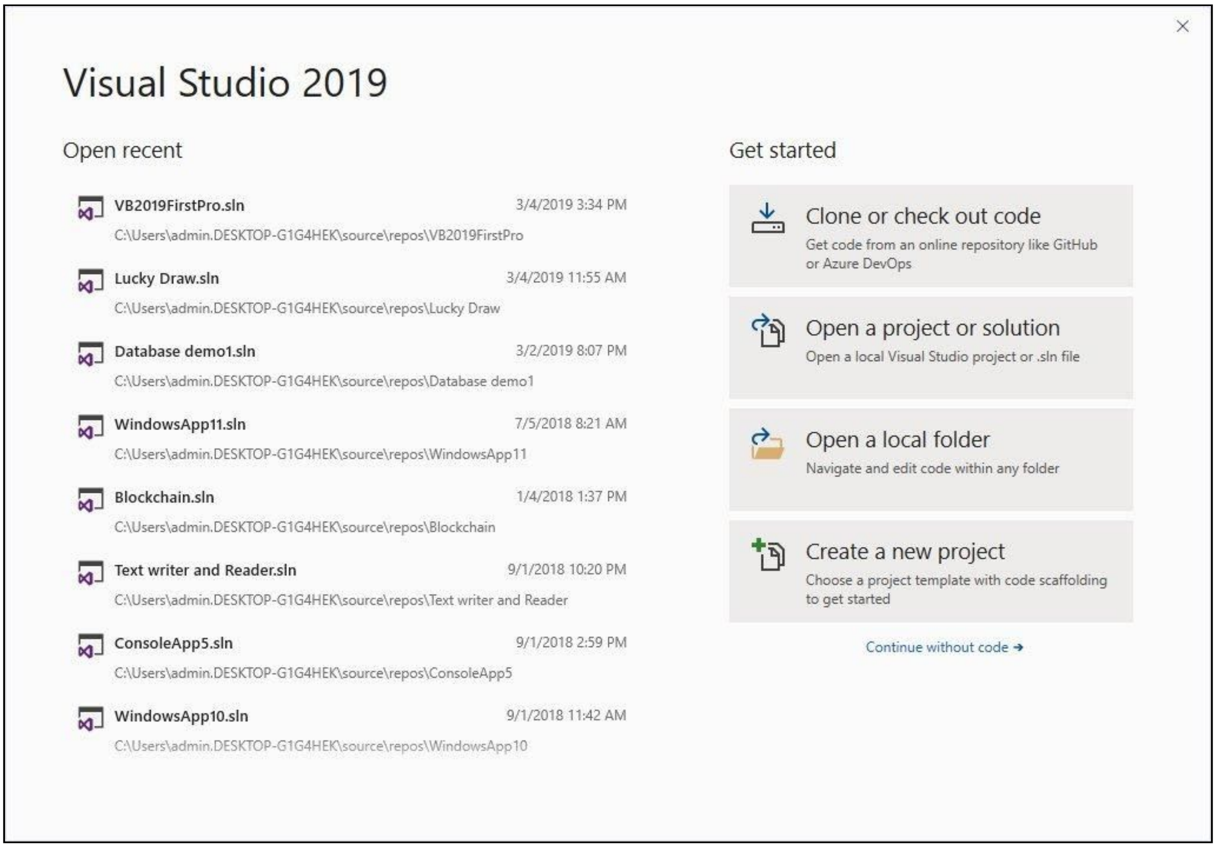
- ☑ .NET profiling tools
- ☑ IntelliCode
- ☑ Text Template Transformation
- ☑ Developer Analytics tools
- □ Modeling SDK
- □ .NET Compiler Platform SDK

Figure 1.3

Upon completion of the installation, you are now ready to launch Visual Studio 2019 and start programming in Visual Basic 2019

1.3 Creating a Visual Basic 2019 Project

To create a Visual Basic 2019 project, launch Microsoft Visual Studio 2019. Launching VS 2019 will bring up the Visual Studio 2019 Start Page, as shown in Figure 1.4



Visual Studio 2019

Open recent

VB2019FirstPro.sln 3/4/2019 3:34 PM
C:\Users\admin.DESKTOP-G1G4HEK\source\repos\VB2019FirstPro

Lucky Draw.sln 3/4/2019 11:55 AM
C:\Users\admin.DESKTOP-G1G4HEK\source\repos\Lucky Draw

Database demo1.sln 3/2/2019 8:07 PM
C:\Users\admin.DESKTOP-G1G4HEK\source\repos\Database demo1

WindowsApp11.sln 7/5/2018 8:21 AM
C:\Users\admin.DESKTOP-G1G4HEK\source\repos\WindowsApp11

Blockchain.sln 1/4/2018 1:37 PM
C:\Users\admin.DESKTOP-G1G4HEK\source\repos\Blockchain

Text writer and Reader.sln 9/1/2018 10:20 PM
C:\Users\admin.DESKTOP-G1G4HEK\source\repos\Text writer and Reader

ConsoleApp5.sln 9/1/2018 2:59 PM
C:\Users\admin.DESKTOP-G1G4HEK\source\repos\ConsoleApp5

WindowsApp10.sln 9/1/2018 11:42 AM
C:\Users\admin.DESKTOP-G1G4HEK\source\repos\WindowsApp10

Get started

Clone or check out code
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

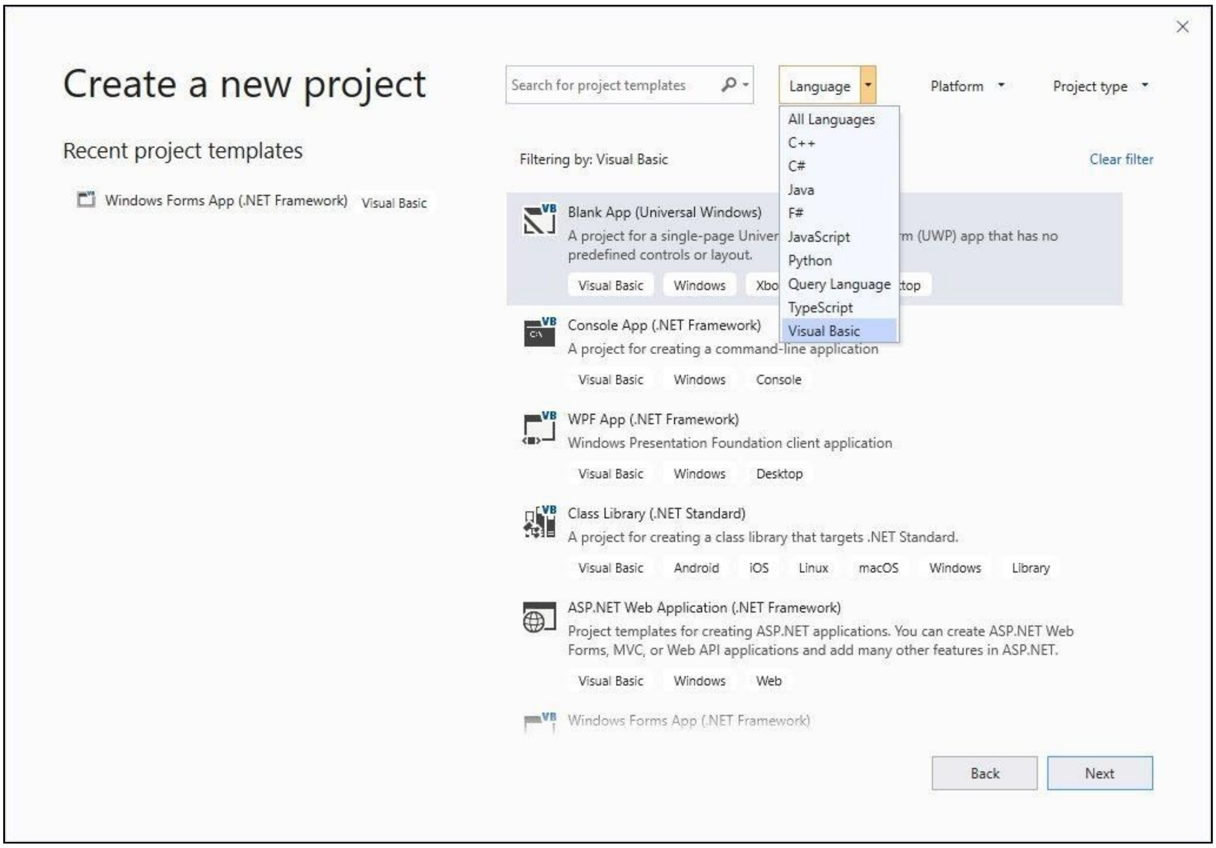
Create a new project
Choose a project template with code scaffolding to get started

Continue without code →

Figure 1.4 Visual Studio 2019 Start Page

The Visual Studio 2019 start page comprises two sections, the Open Recent section and the Get Started section. In the start page, you can select a recent project file or choose any option in the Get Started section. You can choose to clone a project from GitHub or Azure DevOps, open a project or solution, open a local folder, create a new project, or continue without code.

Let us create a new project by clicking on the Create a new project option. You will see the Create a new project template page, as shown in Figure 1.5. In the Create a new project page, select Visual Basic language.



Create a new project

Recent project templates

Windows Forms App (.NET Framework) Visual Basic

Filtering by: Visual Basic

Blank App (Universal Windows)
A project for a single-page Univer... ...m (UWP) app that has no predefined controls or layout.

Console App (.NET Framework)
A project for creating a command-line application

WPF App (.NET Framework)
Windows Presentation Foundation client application

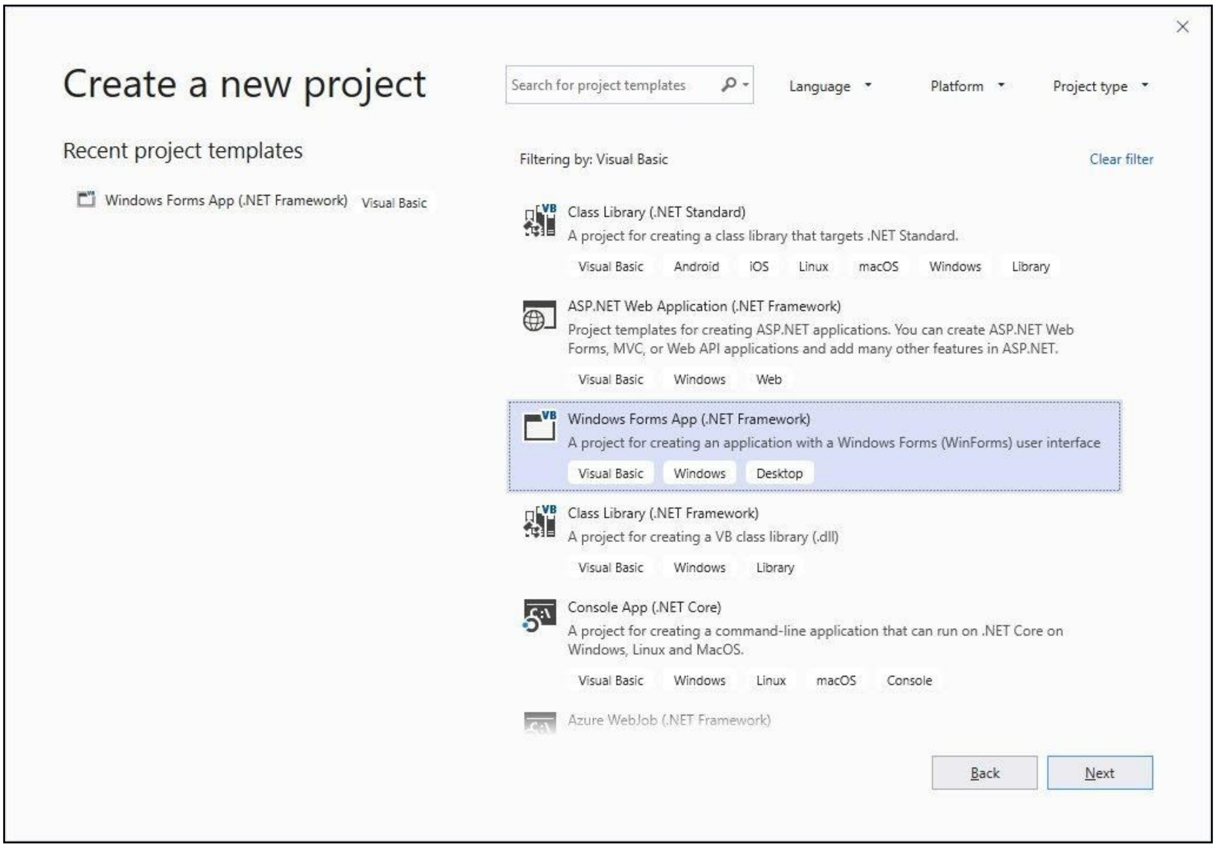
Class Library (.NET Standard)
A project for creating a class library that targets .NET Standard.

ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Windows Forms App (.NET Framework)

Figure 1.5 Create a new project template

Next, select the Windows Forms App (.Net Framework) template as we want to develop a Windows desktop project, as shown in Figure 1.5



Create a new project

Recent project templates

Windows Forms App (.NET Framework) Visual Basic

Filtering by: Visual Basic

Class Library (.NET Standard)
A project for creating a class library that targets .NET Standard.
Visual Basic Android iOS Linux macOS Windows Library

ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
Visual Basic Windows Web

Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
Visual Basic Windows Desktop

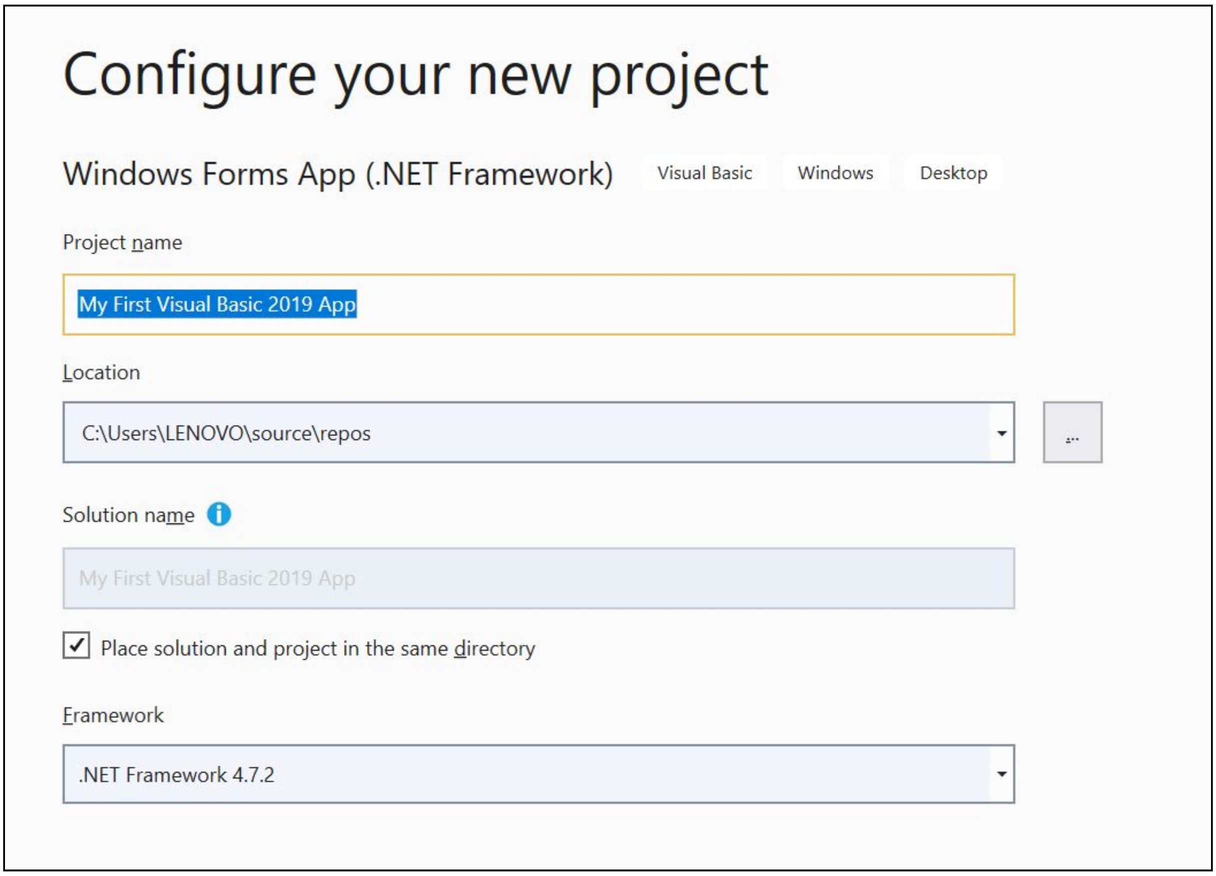
Class Library (.NET Framework)
A project for creating a VB class library (.dll)
Visual Basic Windows Library

Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
Visual Basic Windows Linux macOS Console

Azure WebJob (.NET Framework)

Figure 1.6 Create a new project template

Upon clicking the selected project template, the project configuration page appears, as shown in Figure 1.7. You can configure your project by typing the project name and select a few other options.



Configure your new project

Windows Forms App (.NET Framework)

Project name

My First Visual Basic 2019 App

Location

C:\Users\LENOVO\source\repos

Solution name

My First Visual Basic 2019 App

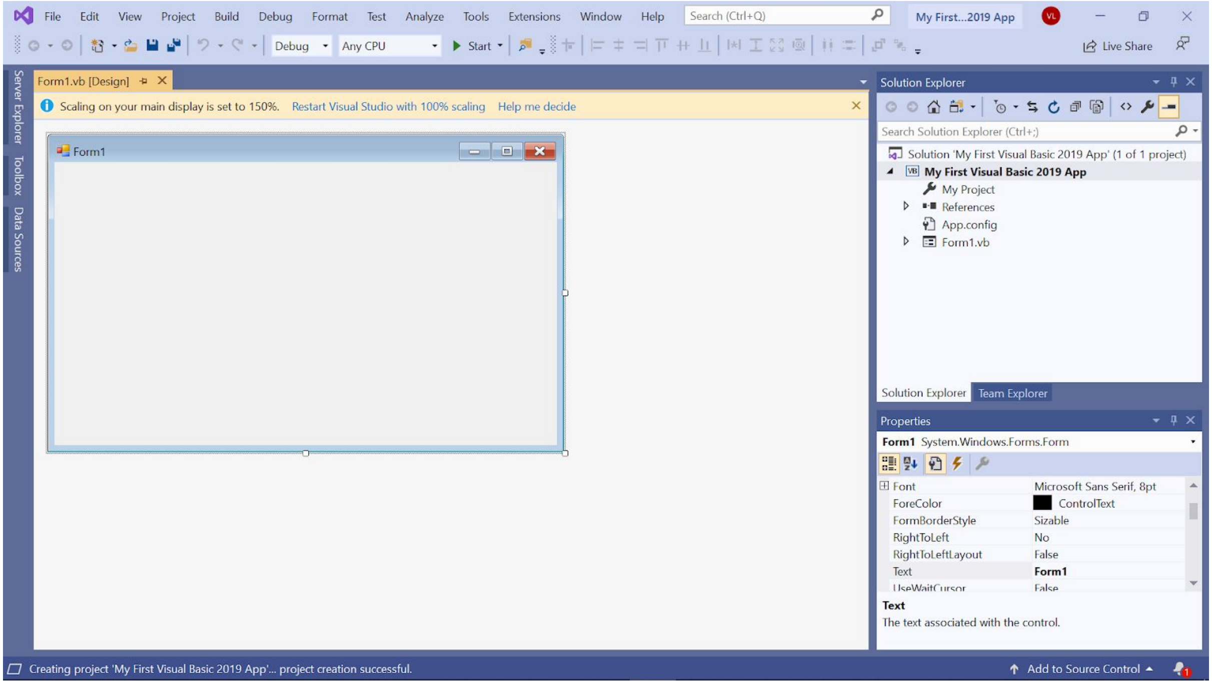
☑ Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Figure 1.7 Configuring Project

At the bottom of this dialog box, you can change the default project name WindowsApplication1 to some other name you like, for example, My First Visual Basic 2019 App. After renaming the project, click OK to continue. The Visual Basic 2019 IDE Windows will appear, as shown in Figure 1.8. Visual Basic 2019 IDE comprises a few windows, the Form window, the Solution Explorer window, and the Properties window. It also consists of a toolbox which contains many useful controls that allows the programmer to develop his or her Visual Basic 2019 programs.



Scaling on your main display is set to 150%. Restart Visual Studio with 100% scaling Help me decide

Form1 System.Windows.Forms.Form

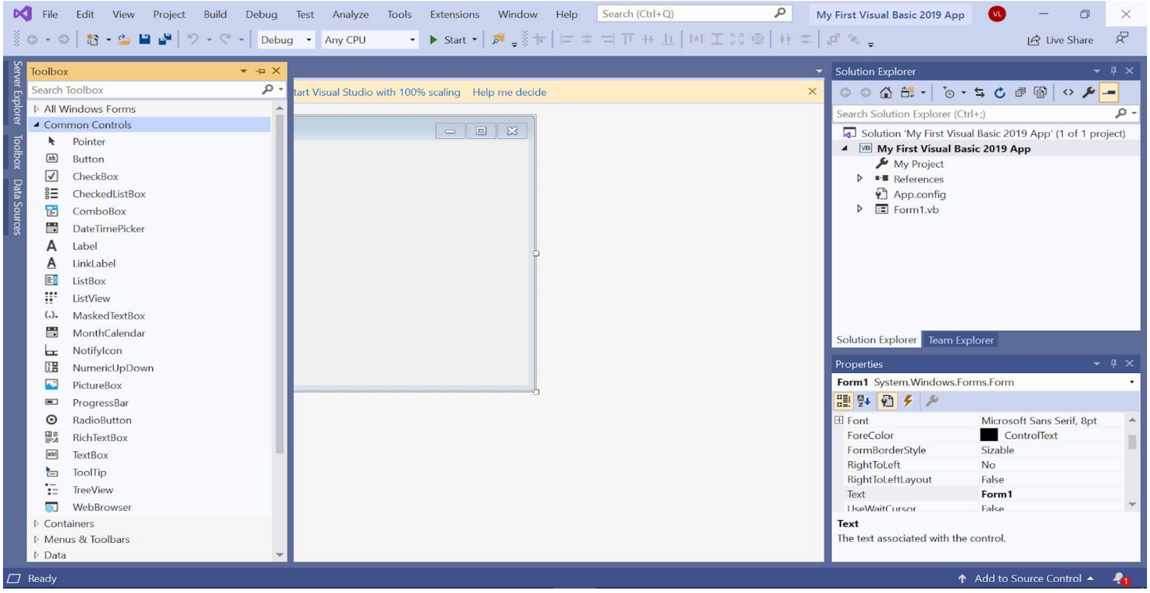
Font	Microsoft Sans Serif, 8pt
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	Form1
UseWaitCursor	False

Text
The text associated with the control.

Creating project 'My First Visual Basic 2019 App'... project creation successful.

Figure 1.8 The Visual Basic 2019 Express IDE

The Toolbox is not shown until you click on the Toolbox tab. When you click on the Toolbox tab or use the shortcut keys Ctrl+Alt+x, the common controls Toolbox will appear, as shown in Figure 1.9. You can drag and move your toolbox around and dock it to the right, left, top or bottom of the IDE.



tart Visual Studio with 100% scaling Help me decide

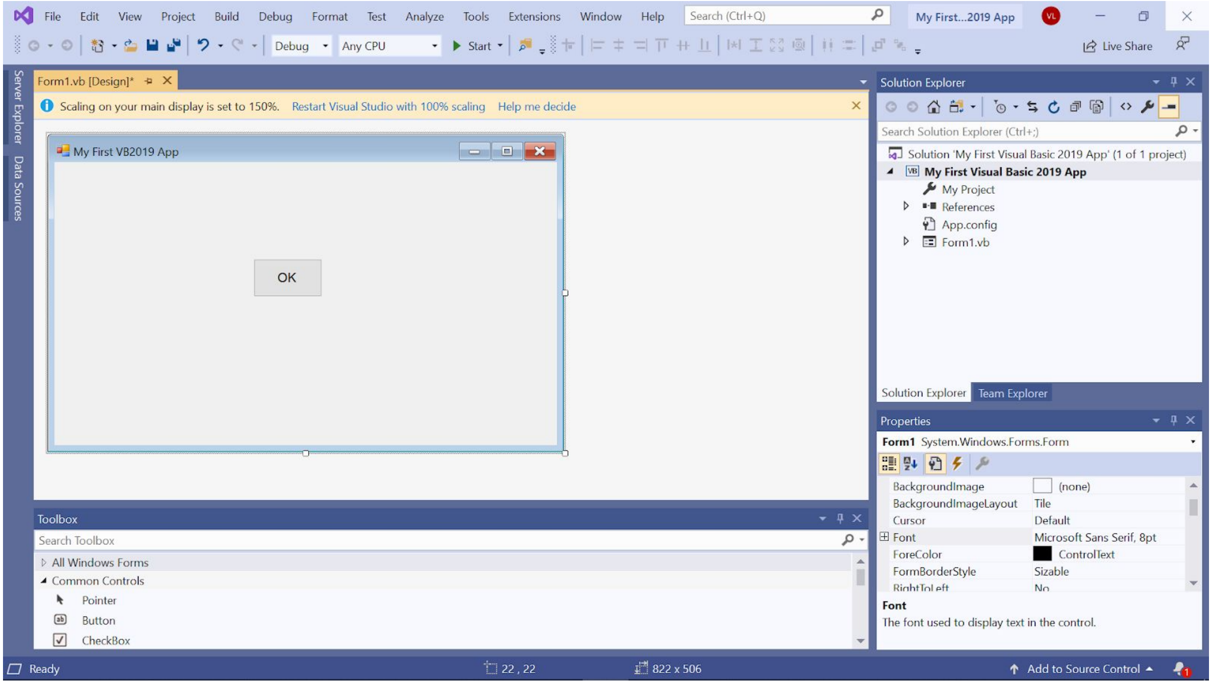
Form1 System.Windows.Forms.Form

Font	Microsoft Sans Serif, 8pt
ForeColor	ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	Form1
UseWaitCursor	False

Text
The text associated with the control.

Figure 1.9 Visual Basic 2019 Toolbox

Next, we shall proceed to show you how to create your first VB2019 application. First, change the text of the form to ‘My First VB 2019 App’ in the properties window; it will appear as the title of the application. Next, insert a button and change its text to OK. The design interface is shown in Figure 1.10



Scaling on your main display is set to 150%. Restart Visual Studio with 100% scaling Help me decide

Solution 'My First Visual Basic 2019 App' (1 of 1 project)
My First Visual Basic 2019 App
My Project
References
App.config
Form1.vb

Form1 System.Windows.Forms.Form

```
BackgroundImage          (none)
BackgroundImageLayout    Tile
Cursor                   Default
Font                     Microsoft Sans Serif, 8pt
ForeColor                ControlText
FormBorderStyle          Sizable
RightToLeft              No
```

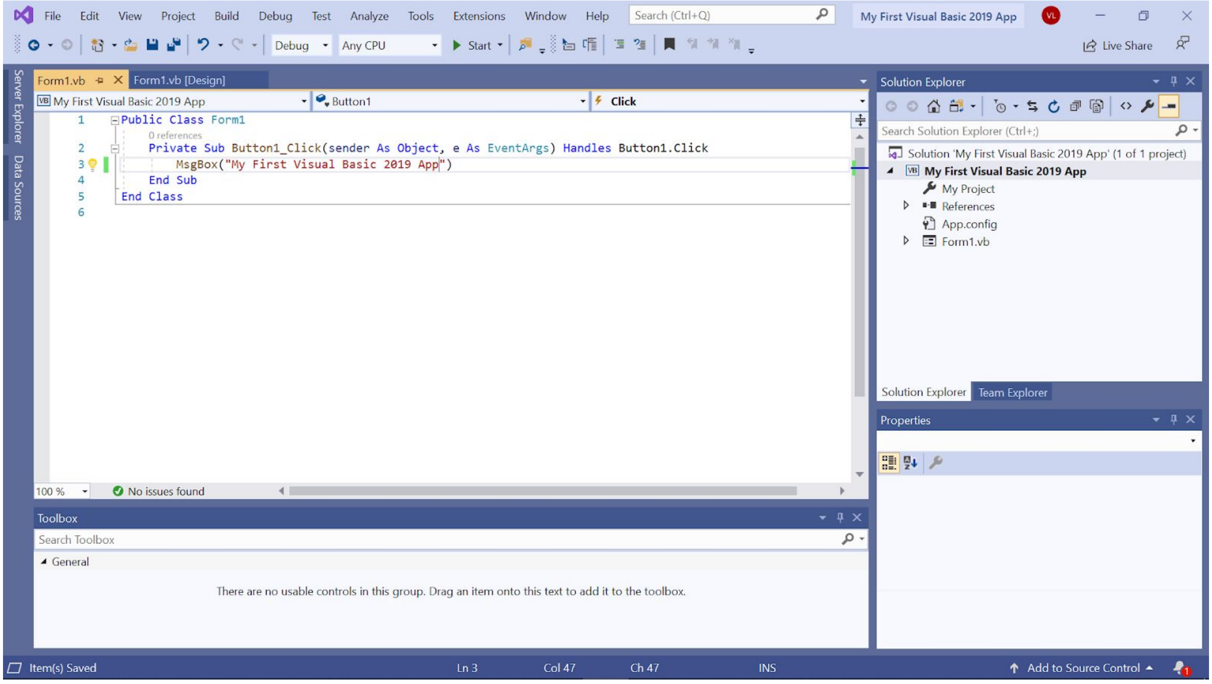
Font
The font used to display text in the control.

Figure 1.10 The Design Interface

Now click on the OK button to bring up the code window and enter the following statement between `Private Sub` and `End Sub` procedure, as shown in Figure 1.11.

```
MsgBox("My First Visual Basic 2019 App")
```

Clicking the Start button on the toolbar or press F5 to run the application will launch the runtime interface, as shown in Figure 1.12. Executing the application by clicking on the OK button will bring up a dialog box that displays the “My First Visual Basic 2019 App" message, as shown in Figure 1.13. The function `MsgBox` is a built-in function of Visual Basic 2019 which can display the text enclosed within the brackets.



```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        MsgBox("My First Visual Basic 2019 App")
    End Sub
End Class
```

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

Figure 1.11 Visual Basic 2019 Code Window

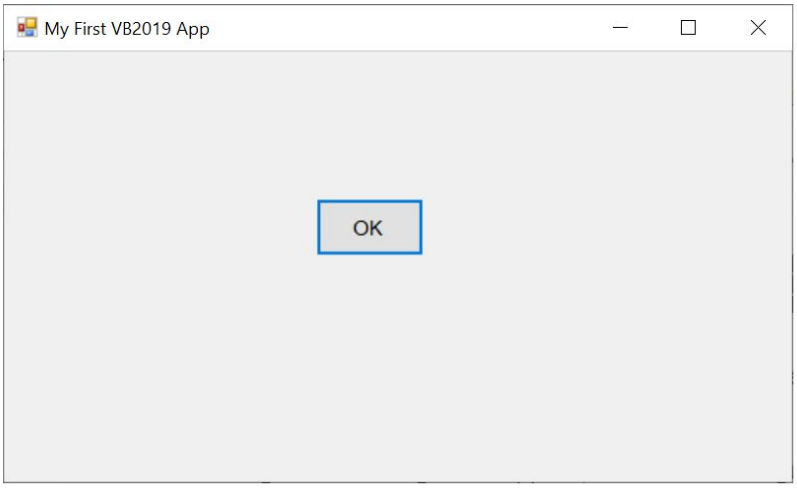
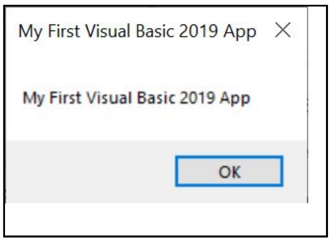


Figure 1.12 The Runtime Interface



My First Visual Basic 2019 App

My First Visual Basic 2019 App

Figure 1.13 The Message Box

Summary

- In section 1.1, you have learned about the history of Visual Basic 2019
- In section 1.2, you have learned how to install and launch Visual Basic Studio 2019
- In section 1.3, you have learned how to launch the new project dialog and the Visual Basic 2019 IDE. You have also learned how to write your first program.

Chapter 2 Designing the User Interface

The first step in developing an application is to design the user interface (UI). To build a graphical user interface, first you need to customize the default form by changing its properties at design phase and at runtime, including its name, title, background color and so forth. After customizing the default form, you may proceed to add controls from the toolbox to the form and then customize their properties.

2.1 Customizing the Form

When you start a new Visual Basic 2019 project, the VB2019 IDE will display the default form along with the Solution Explorer window and the Properties window, as shown in Figure 2.1. The name of the default form is Form1. The properties window displays all the properties associate with Form1 and their corresponding attributes or values. You can change the name of the form, the title of the form using the text property, the background color, the foreground color, size and more. Try changing the properties as shown in Table 2.1

Table 2.1

Property	**Value**
Name	MyForm
Text	My First VB2019 App
BackColor	LavenderBlush
ForeColor	Crimson
MaximizeBox	False

You do not have to type in the color manually, you can indeed select a color from the color drop-down list that comprises three tabs, Custom, Web, and System, as shown in Figure 2.1. Clicking on the drop-down arrow will bring out a color palette or a list of color rectangles where you can select a color.

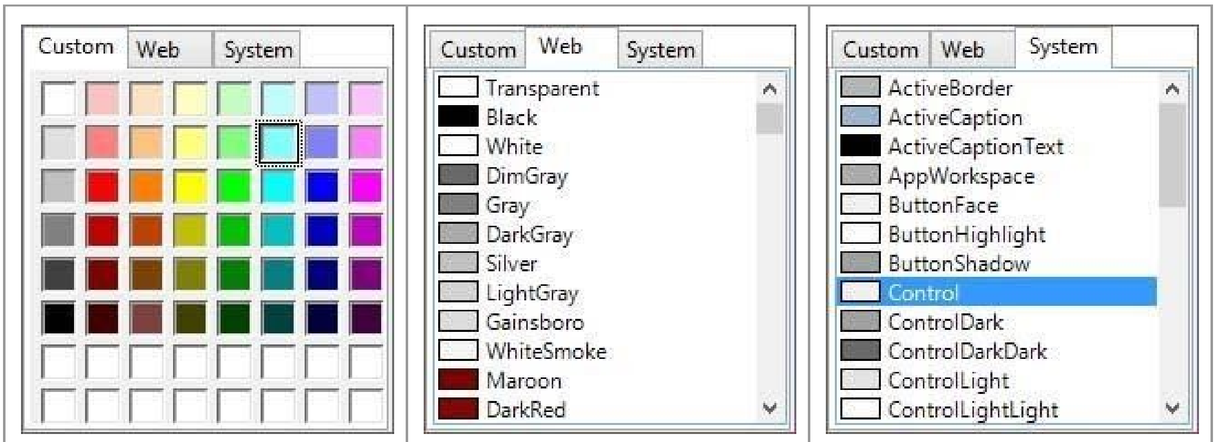
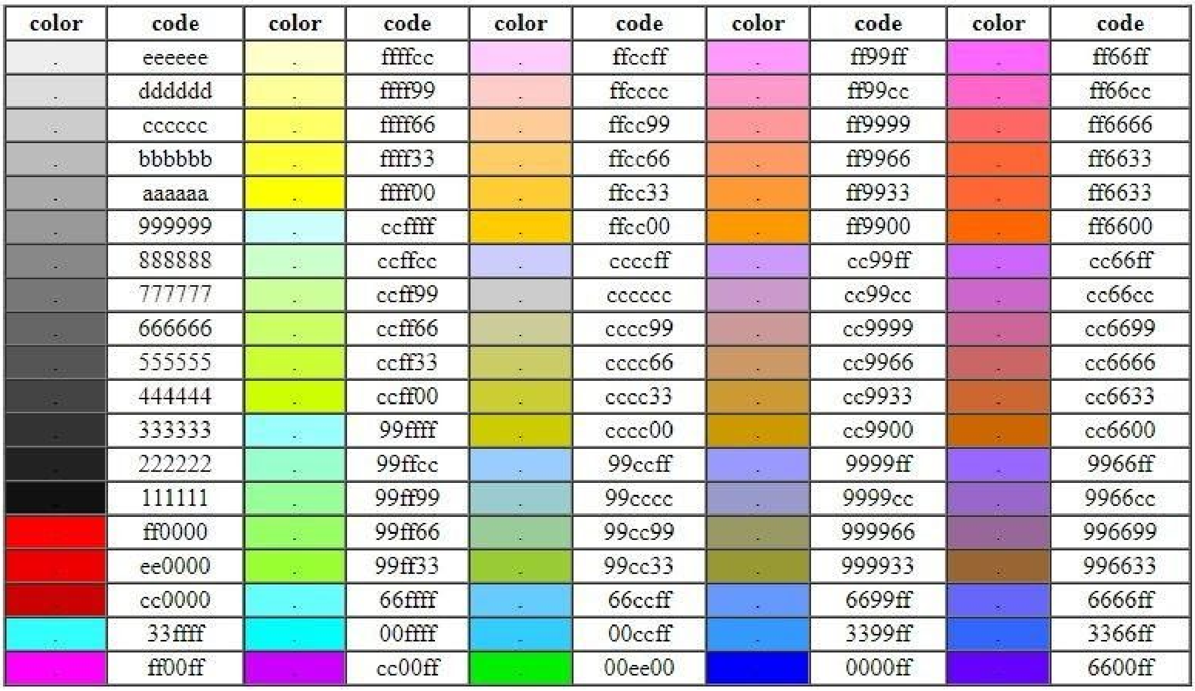


Figure 2.1

Another method of setting the colors is to manually type in the RGB color code or the hex color code. The values of R, G and B ranges from 0 to 255, therefore, by varying the values of the RGB we can obtain different colors.

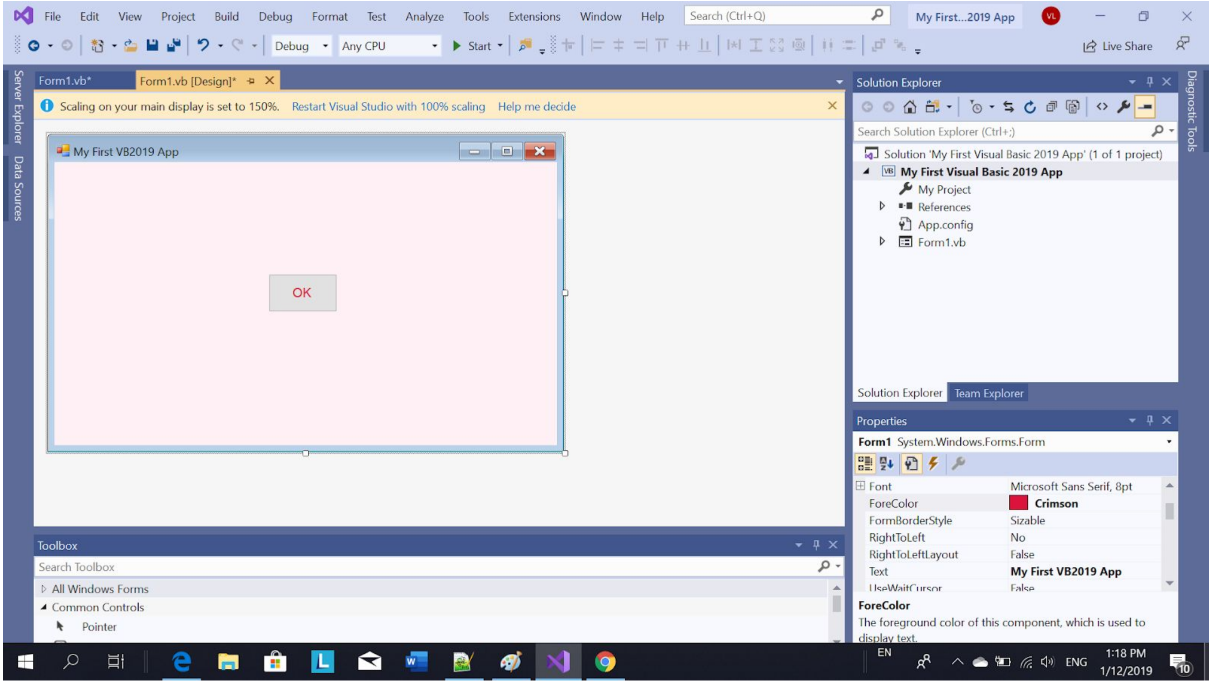
The hex color code system uses a six-digit, three-byte hexadecimal number to represent colors. The bytes represent the red, green and blue components of the color. One byte represents a number ranging from 00 to ff, or 0 to 255 in decimal notation. For example, `#0000ff` represents the cyan color. However, when you type the Hex color code in the properties window, it automatically converts the color to RGB color. Figure 2.2 shows a list of Hex color codes and the corresponding colors.



color	code	color	code	color	code	color	code	color	code
.	eeeeee	.	ffffcc	.	ffccff	.	ff99ff	.	ff66ff
.	dddddd	.	ffff99	.	ffcccc	.	ff99cc	.	ff66cc
.	cccccc	.	ffff66	.	ffcc99	.	ff9999	.	ff6666
.	bbbbbb	.	ffff33	.	ffcc66	.	ff9966	.	ff6633
.	aaaaaa	.	ffff00	.	ffcc33	.	ff9933	.	ff6633
.	999999	.	ccffff	.	ffcc00	.	ff9900	.	ff6600
.	888888	.	ccffcc	.	ccccff	.	cc99ff	.	cc66ff
.	777777	.	ccff99	.	cccccc	.	cc99cc	.	cc66cc
.	666666	.	ccff66	.	cccc99	.	cc9999	.	cc6699
.	555555	.	ccff33	.	cccc66	.	cc9966	.	cc6666
.	444444	.	ccff00	.	cccc33	.	cc9933	.	cc6633
.	333333	.	99ffff	.	cccc00	.	cc9900	.	cc6600
.	222222	.	99ffcc	.	99ccff	.	9999ff	.	9966ff
.	111111	.	99ff99	.	99cccc	.	9999cc	.	9966cc
	ff0000	.	99ff66	.	99cc99	.	999966	.	996699
	ee0000	.	99ff33	.	99cc33	.	999933	.	996633
	cc0000	.	66ffff	.	66ccff	.	6699ff	.	6666ff
.	33ffff		00ffff	.	00ccff	.	3399ff	.	3366ff
	ff00ff		cc00ff		00ee00		0000ff		6600ff

Figure 2.2 Hex Color Codes

The design interface is shown in Figure 2.3 and the runtime interface is shown in Figure 2.4. In the runtime interface, notice that the title has been changed from Form1 to My First Visual Basic 2019 App, background color changed to Lavender Blush , the text OK color is Crimson and the window cannot be maximized.



Form1 System.Windows.Forms.Form

Font	Microsoft Sans Serif, 8pt
ForeColor	🟥 Crimson
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	My First VB2019 App
UseWaitCursor	False

ForeColor
The foreground color of this component, which is used to display text.

Figure 2.3 Design UI

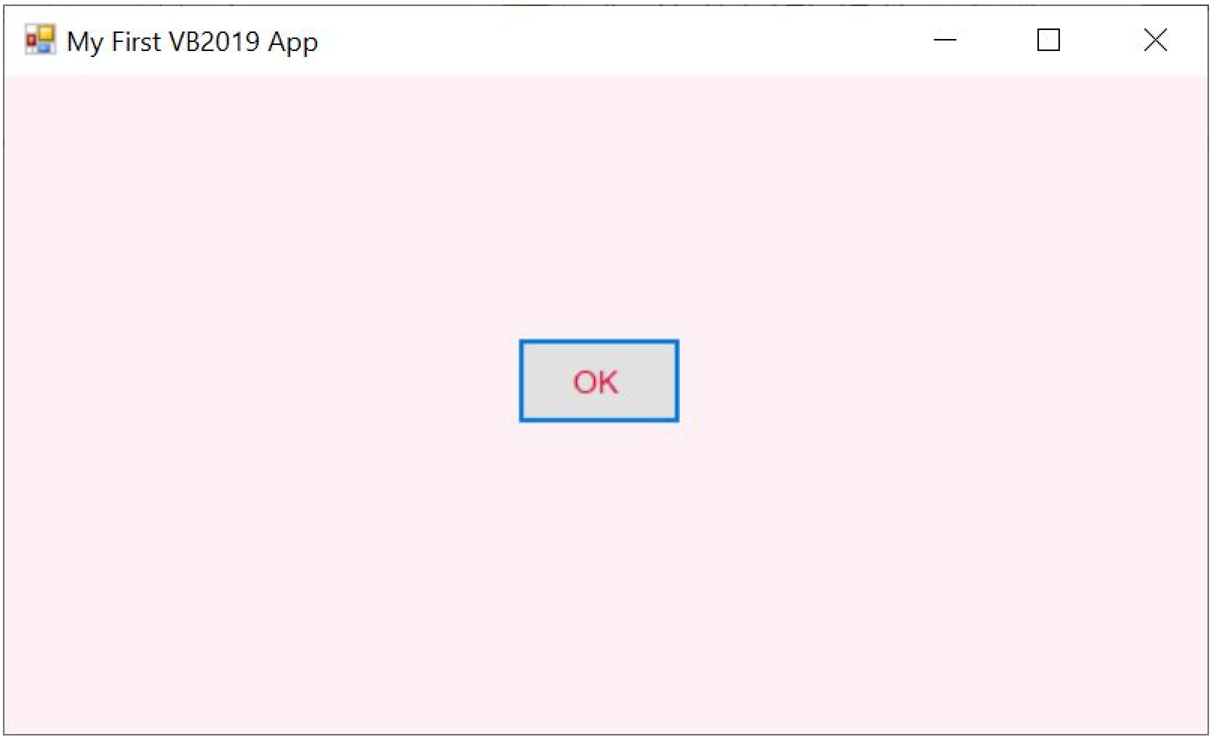


Figure 2.4 Runtime UI

You can also change the properties of the form at runtime by writing relevant codes. The default form is an object and an instant of the form can be denoted by the name `Me`. The property of the object can be defined by specifying the object's name followed by a period, as follows:

```
ObjectName.property
```

For example, setting the background of the form to blue using the following code:

```
Me.BackColor=Color.Blue
```

In addition, you can use the `FromArgb` method to specify the color using the RGB codes, as follows:

```
Me.BackColor = Color.FromArgb(0, 255, 0)
```

Example 2.1 Changing Properties at Runtime

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles_
MyBase.Load
 Me.Text = "My First Visual Basic 2019 Application"
 Me.BackColor = Color.Turquoise
 Me.ForeColor = Color.Ivory
 MyBtn.BackColor = Color.DodgerBlue
 Me.MaximizeBox = False
 Me.MinimizeBox = True
End Sub
```

To runtime UI is shown in Figure 2.5. Notice that is now different from that shown in Figure 2.4,

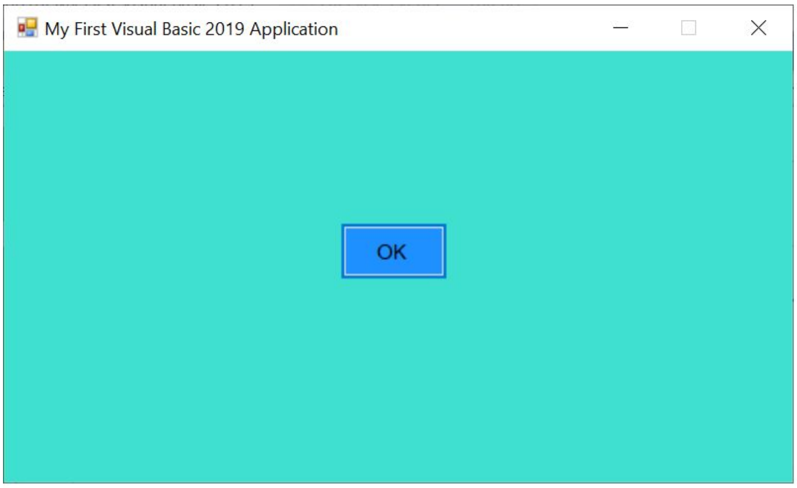


Figure 2.5

In place of `Turquoise`, you can use RGB code as follows:

```
Me.BackColor = Color.FromArgb(64,224,208)
```

In addition, you can specify the size, the opacity and the position of the default form using the code, as in Example 2.2

Example 2.2 Customizing the Form

```
Private Sub Form1_Load(sender As Object, e As EventArgs Handles
MyBase.Load

      Me.Text = "My First VB2019 App"
      Me.BackColor =Color.Beige
      Me.MaximizeBox = False
```

```
        Me.MinimizeBox = True
        Me.Size = New Size(400, 400)
        Me.Opacity = 0.85
        Me.CenterToParent()

End Sub
```

The property Opacity sets the degree of transparency. The runtime interface is as shown in Figure 2.6

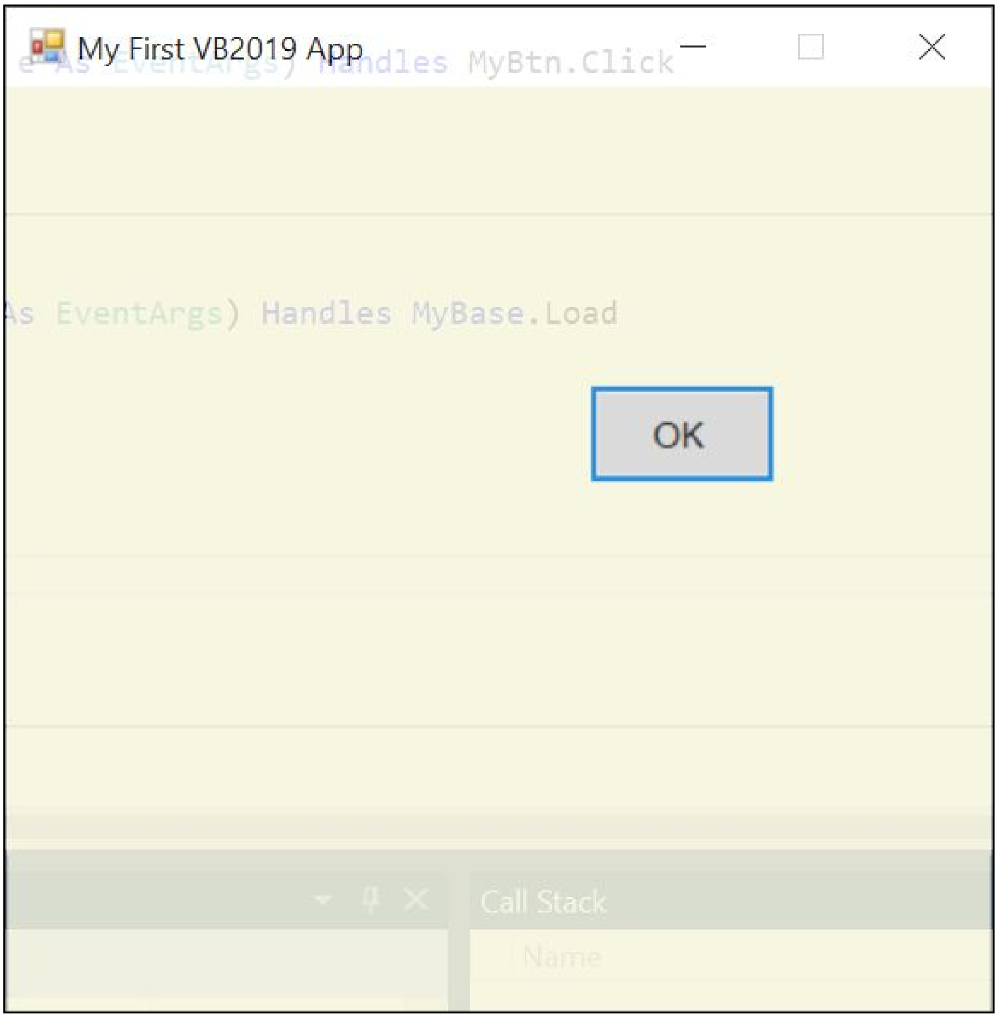
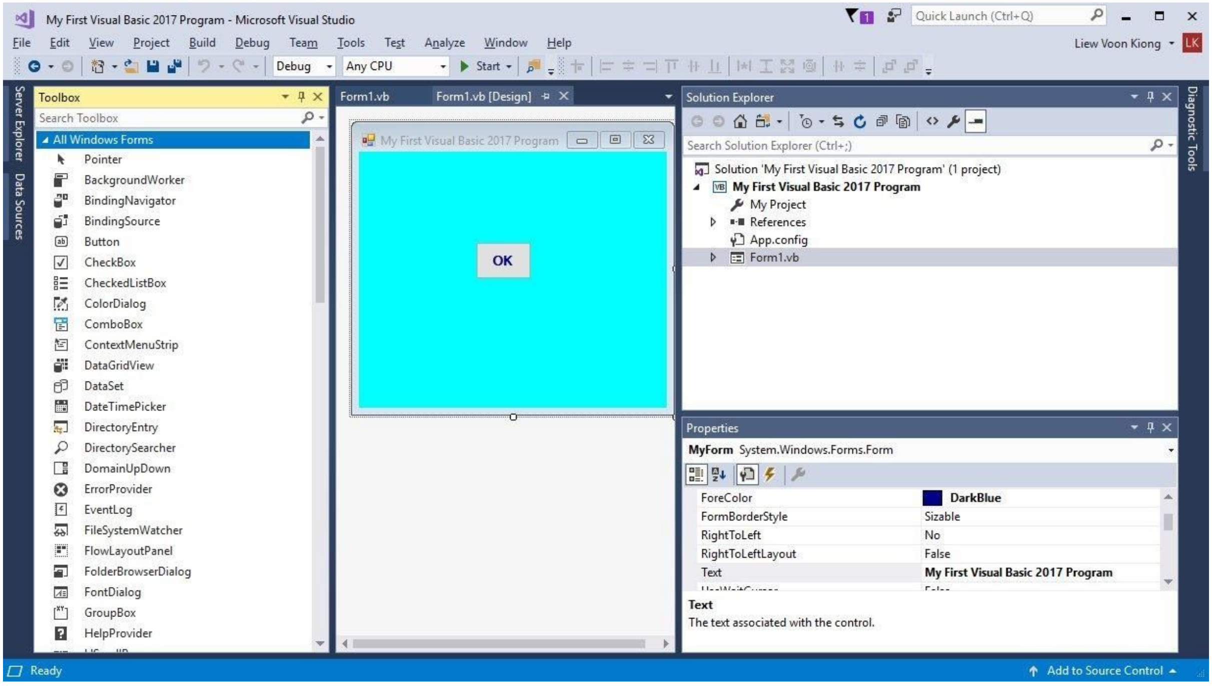

My First VB2019 App
e As EventArgs) Handles MyBtn.Click
As EventArgs) Handles MyBase.Load
OK
Call Stack
Name

Figure 2.6

2.2 Adding Controls to the Form

In section 2.1, we have learned how to build an UI by customizing the default form. Next, we shall continue to build the UI by adding some controls to the form. The controls are objects that consist of three elements, namely properties, methods, and events. We can add them to the form from the Toolbox. Among the controls, the most used ones are button, label, textbox, list box, combo box, picture box, check box, radio button and more. These controls can be made visible or invisible at runtime. However, some controls will only run in the background and never be seen at runtime, one such control is the timer.

The Toolbox is usually hidden when you start Visual Basic 2019 IDE, you need to click View on the menu bar and then select Toolbox to reveal the toolbox, as shown in Figure 2.6. You can also use shortcut keys Ctrl+w+x to bring out the toolbox.



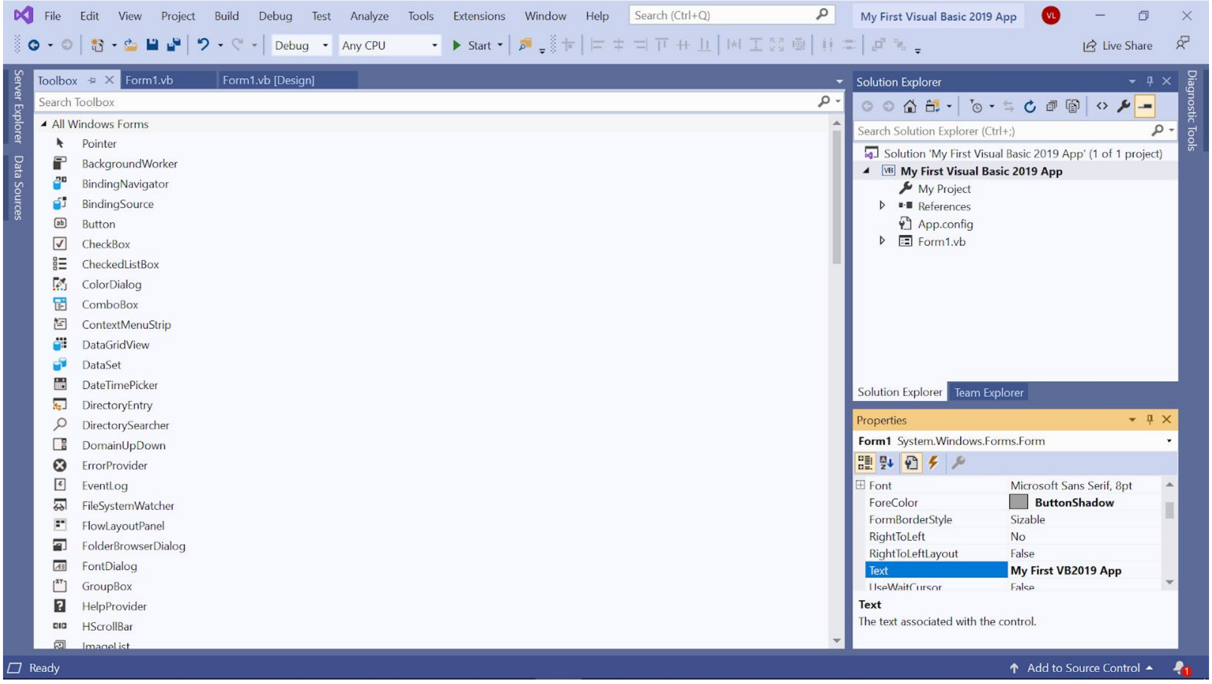
MyForm System.Windows.Forms.Form

ForeColor	DarkBlue
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	My First Visual Basic 2017 Program

Text
The text associated with the control.

Figure 2.6: Toolbox

You can position the Toolbox by dragging it anywhere you like while its status is set to float. You can also dock the toolbox by right-clicking on the toolbox and choose dock from the pop-up menu. The docked Toolbox appears side by side with the Solution Explorer, or as one of the tabbed windows together with the Form Design window and the code window, as shown in Figure 2.7.



Form1 System.Windows.Forms.Form

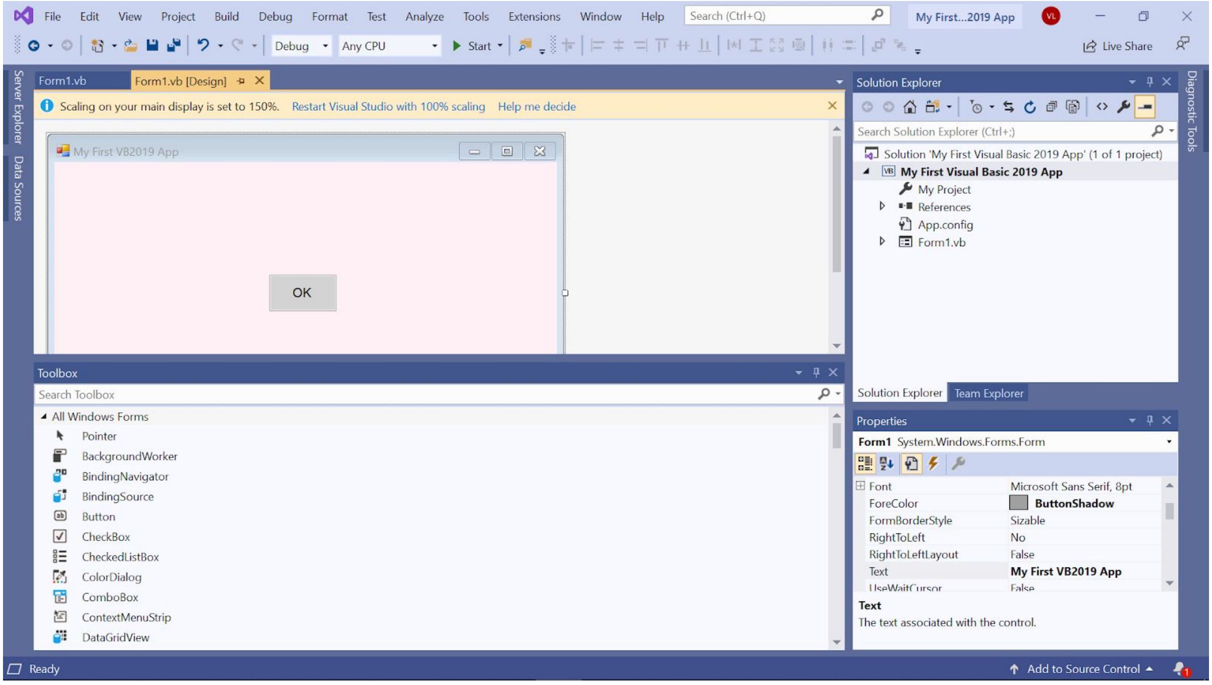
Font	Microsoft Sans Serif, 8pt
ForeColor	ButtonShadow
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	My First VB2019 App
UseWaitCursor	False

Text
The text associated with the control.

Figure 2.7 Toolbox

You can also dock the toolbox at the bottom, below the default form, as shown in Figure 2.8. Further, you may also pin the toolbox to the side bar or the bottom bar by clicking on the pin icon on the menu bar of the toolbox.

How and where you want to position your toolbox is entirely up to you but we strongly suggest that you place the tool box alongside or at the bottom of the default form for ease of use. You should never cover the form with the toolbox because it will be difficult to add controls to the form.



Scaling on your main display is set to 150%.

Font	Microsoft Sans Serif, 8pt
ForeColor	ButtonShadow
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	My First VB2019 App
UseWaitCursor	False

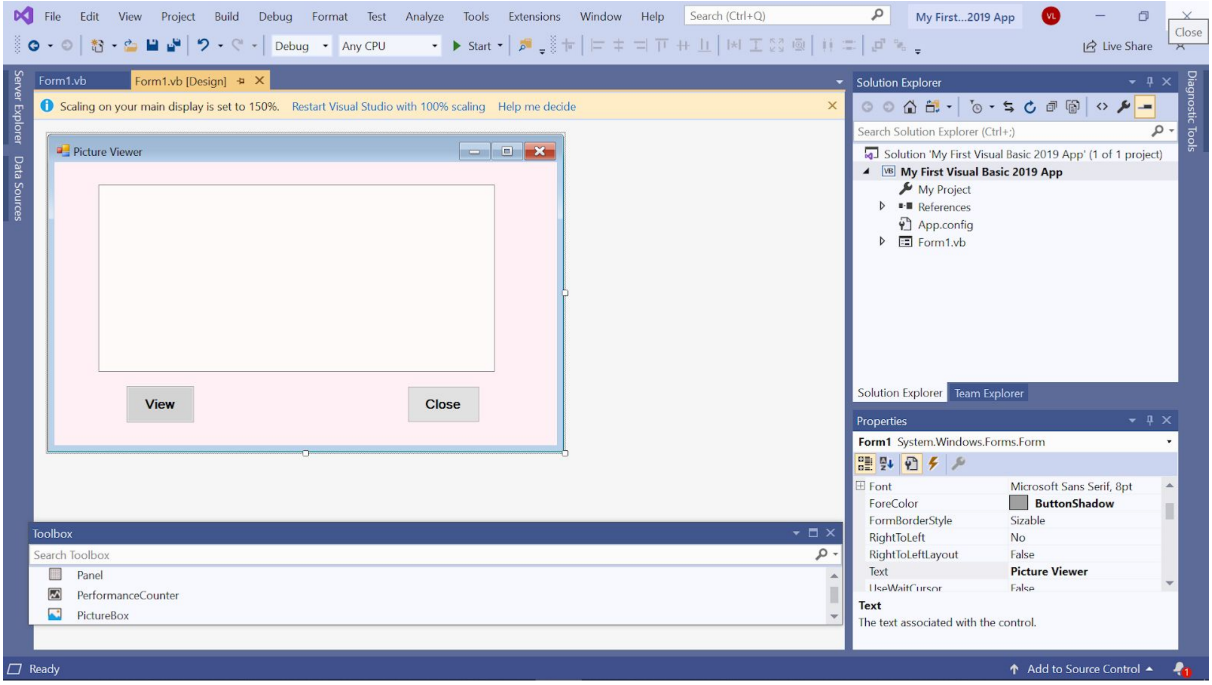
Text
The text associated with the control.

Figure 2.8

Adding a control to the form is an easy task, what you need to do is double click it or drag it onto the form. You can drag the control around the form, and resize it.

To demonstrate how to add the controls and then change their properties, we shall design a picture viewer. First, change the title of the default form to Picture Viewer in its properties window. Next, insert a picture box on the form and change its background color to white. To do this, right click the picture box and select properties in the popup menu, then look for the `BackColor` Property as shown in the properties window in Figure 2.9.

Finally, add two buttons to the form and change the text to View and Close in their respective properties' windows. The picture viewer is not functional yet until we write code for responding to events triggered by the user. We will deal with the programming part in the coming chapters.



Scaling on your main display is set to 150%.

Form1 System.Windows.Forms.Form

Font	Microsoft Sans Serif, 8pt
ForeColor	ButtonShadow
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	Picture Viewer
UseWaitCursor	False

Text
The text associated with the control.

Figure 2.9

Summary

- In section 2.1, you have learned how to customize the form by changing the values of its properties.
- In section 2.2, you have learned how to add controls to the form and change their properties at design phase and at runtime.

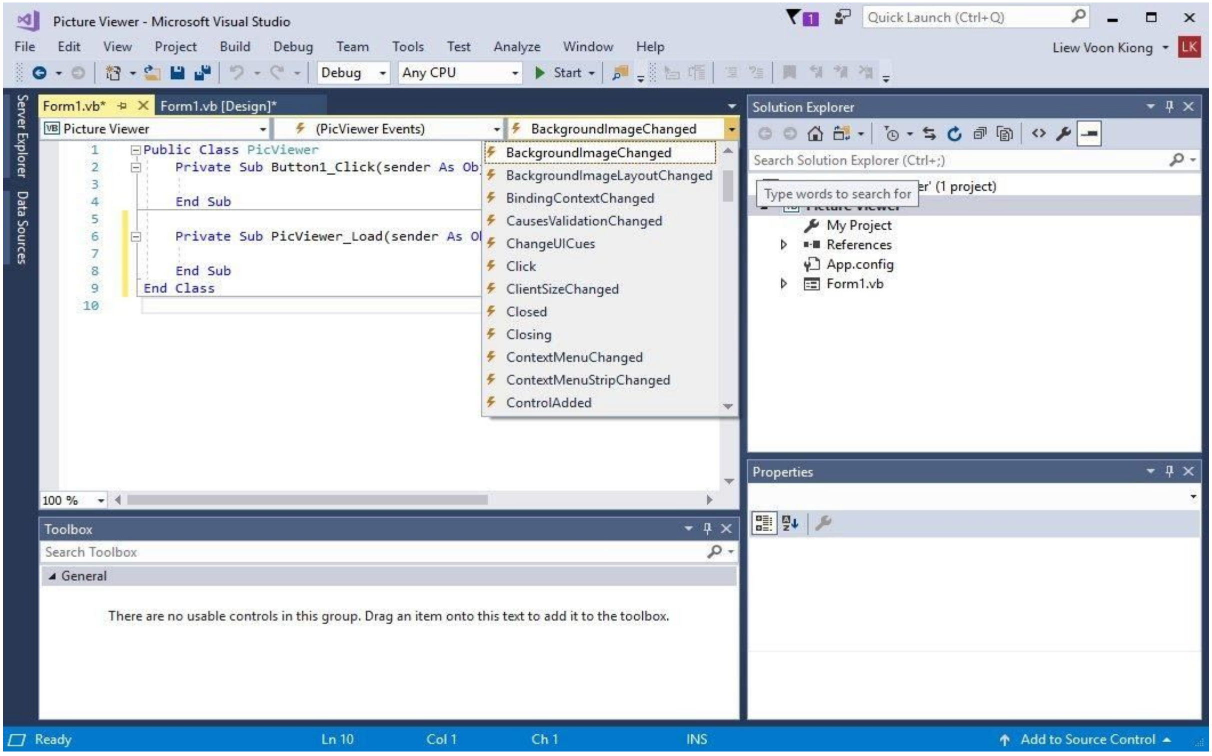
Chapter 3 Writing the Code

In the previous chapter, we have learned how to design the user interface by adding controls to the form and by changing their properties. However, the user interface alone will not work without adding code to them. In this chapter, we shall learn how to write code for all the controls to enable them to interact with the events triggered by the users. Before learning how to write Visual Basic 2019 code, let us delve into the concept of event-driven programming

3.1 The Concept of Event-Driven Programming

Visual Basic 2019 is an event-driven programming language, meaning that the code is executed in response to events triggered by the user like clicking the mouse or pressing a key on the keyboard. Some other events are selecting an item from a drop-down list, typing some words into a text box and more. It may also be an event that runs in response to some other program. Some of the common events in Visual Basic 2019 are load, click, double-click, drag-drop, keypress and more.

Every control you place on the form has a set of events associate with it. To view the events, double-click the control on the form to enter the code window. The default event will appear at the top right side of the code window. You must click on the default event to view other events associated with the control. The code appears on the left side is the event procedure associated with the load event. Figure 3.1 illustrates the event procedure Load associated with the Form (its name has been changed to PicViewer therefore you can see the words PicViewer events) and Figure 3.2 shows the events associated with the button.



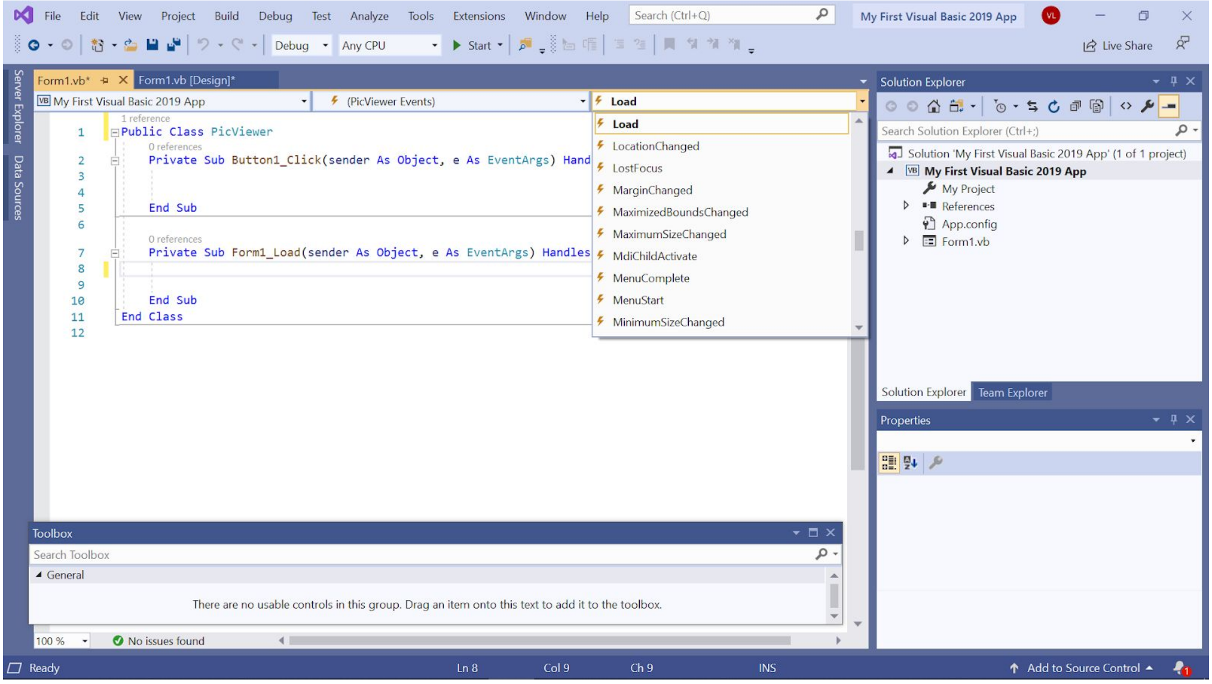
```
Public Class PicViewer
    Private Sub Button1_Click(sender As Ob

    End Sub

    Private Sub PicViewer_Load(sender As Ol

    End Sub
End Class
```

Figure 3.1: Events associated with Form



```
Public Class PicViewer
    Private Sub Button1_Click(sender As Object, e As EventArgs) Hand


    End Sub

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles


    End Sub
End Class
```

Figure 3.2: Events associated with the button

3.2 Writing the Code

To write code, click on any part of the form to go into the code window, as shown in Figure 3.1. The event procedure is to load Form1 which starts with the keywords `Private Sub` and ends with `End Sub`. This procedure includes the Form1 class and the event Load, and they are bound together with an underscore, i.e. `Form_Load`. It does nothing other than loading an empty form. To make the load event do something, insert the following statement.

```
MsgBox ( "Welcome to Visual Basic 2019")
```

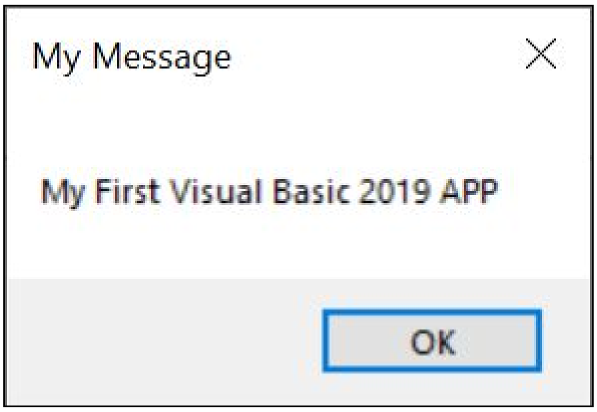
Example 3.1 Displaying a Message

```
Public Class Form1
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles_
MyBase.Load

      MsgBox ( "My First Visual Basic 2019 APP", ,"My Message")

End Sub
End Class
```

MsgBox is a built-in function that displays a message in a pop-up message box. The MsgBox function comprises a few arguments, the first is the message that is displayed and the third one is the title of the message box. When you run the program, a message box displaying the phrase "My First Visual Basic 2019 APP" will appear, as shown in Figure 3.3.



My Message

My First Visual Basic 2019 APP

Figure 3.3

You will notice that above the Private Sub structure there is a preceding keyword `Public Class Form1`. This has to do with the concept of the object-oriented programming language. When we start a windows application in Visual Basic 2019, we will see a default form with the name Form1 appear in the IDE, it is the Form1 Class that inherits from the Form class `System.Windows.Forms.Form`.

Now, let us write a code that perform arithmetic calculation, as in Example 3.2

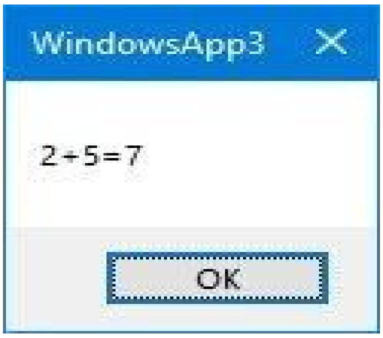
Example 3.3 Arithmetic Calculations

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

      MsgBox("2" & "+" & "5" & "=" & 2 + 5)

End Sub
```

*The symbol & (ampersand) is to perform string concatenation. The output is as shown in Figure 3.4



```
2+5=7
```

Figure 3.4

Summary

- In section 3.1, you have learned the concepts of event driven programming
- In section 3.2, you have learned how to write code for the controls