

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.

Copyright © 2019 Liew Voon Kiong

All rights reserved. No Part of this e-book may be reproduced in any form or by any means, without permission in writing from the author.

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 to Visual Basic 2019	11
1.1 A Brief Description of Visual Basic 2019	11
1.2 Installation of Visual Studio 2019	12
1.3 Creating a Project in Visual Basic 2019	14
Chapter 2 Designing the User Interface	23
2.1 Customizing the Form	23
2.2 Adding Controls to the Form	29
Chapter 3 Writing the Code	34
3.1 The Concept of Event-Driven Programming	34
3.2 Writing the Code	36
Chapter 4 Working with Controls	39
4.1 TextBox	39
Example 4.1	39
4.2 Label	41
Example 4.2	41
4.3 ListBox	42
4.3.1 Adding Items to a Listbox	42
a) Adding items using the String Collection Editor	42
b) Adding Items using the Add() Method	44
Example 4.3	45
Example 4.4	45
Example 4.5 Geometric Progression	47
4.3.2 Removing Items from a List Box	49
Example 4.5	49
Example 4.6	50
Example 4.7	51
Example 4.8	53

Example 4.9	53
4.4 ComboBox	54
4.4.1 Adding Items to a ComboBox	54
4.4.2 Removing Items from a Combobox	59
Chapter 5 Handling Images	61
5.1 Loading an Image in a Picture Box	61
5.1.1 Loading an Image at Design Time	61
5.1.2 Loading an Image at Runtime	64
5.2 Loading an Image in a Picture Box using Open File Dialog Control	64
Chapter 6 Working with Data	68
6.1 Visual Basic 2019 Data Types	68
6.1.1 Numeric Data Types	68
Table 6.1: Numeric Data Types	69
6.1.2 Non-numeric Data Types	69
Table 6.2: Non-numeric Data Types	70
6.1.3 Suffixes for Literals	70
Table 6.3: Suffixes and Data Types	70
6.2 Variables and Constants	71
6.2.1 Variable Names	71
6.2.2 Declaring Variables	72
Example 6.1	72
Example 6.2	73
Example 6.3	73
6.2.3 Assigning Values to Variables	74
Example 6.4	74
6.2.4 Scope of Declaration	75
6.2.5 Declaring Constants	76
Chapter 7 Arrays	78
7.1 Introduction to Arrays	78

7.2 Dimension of an Array	78
7.3 Declaring Arrays	79
Example 7.1	80
Example 7.2	81
Example 7.3	81
Example 7.4	82
Chapter 8 Performing Mathematical Operations	84
8.1 Mathematical Operators	84
8.2 Writing Code that Performed Mathematical Operations	85
Example 8.1 Standard Arithmetic Calculations	85
Example 8.2 Pythagorean Theorem	86
Example 8.3: BMI Calculator	86
Chapter 9 String Manipulation	89
9.1 String Manipulation Using + and & signs	89
Example 9.1	89
Example 9.2	90
9.2 String Manipulation Using Built-in Functions	92
9.2 (a) The Len Function	92
Example 9.3	92
9.2(b) The Right Function	93
Example 9.4	93
9.2(c) The Left Function	93
9.2(d) The Mid Function	94
Example 9.5	94
Example 9.6	94
9.2(e)The Trim Function	96
Example 9.7	96
9.2(f) The Ltrim Function	96
9.2(g)The Rtrim Function	96

9.2(h) The InStr function	97
9.2(i) The Ucase and the Lcase Functions	97
9.2(j) The Chr and the Asc functions	97
Chapter 10 Using If...Then.. Else	
10.1 Conditional Operators	99
10.1 Conditional Operators	99
10.2 Logical Operators	100
10.3 Using If ...Then...Else	101
10.3(a) If ...Then Statement	101
Example 10.1	101
10.3(b) If...Then...Else Statement	101
Example 10.2	102
Example 10.3	105
10.3(c) If ...Then...Elseif Statement	107
Example 10.4 Grade Generator	107
Chapter 11 Using Select Case	110
11.1 The Select Case...End Select Structure	110
11.2 The usage of Select Case is shown in the following examples	111
Example 11.1: Examination Grades	111
Example 11.2	112
Example 11.3	113
Example 11.4	114
Chapter 12 Looping	116
12.1 For ...Next Loop	116
Example 12.1 a	117
Example 12.1b	117
Example 12.1c	117
Example 12.1d	118
12.2 Do Loop	118

Example 12.2(a)	119
Example 12.2(b)	119
12.3 While ...End While Loop	120
Example 12.3	121
Chapter 13 Sub Procedures	122
13.1 What is a Sub Procedure	122
13.2 Examples of Sub Procedure	122
Example 13.1	122
Example 13.2: Password Cracker	113
Chapter 14 Creating Functions	127
14.1 Creating User-Defined Functions	127
Example 14.1: BMI Calculator	128
Example 14.2: Future Value Calculator	129
14.2 Passing Arguments by Value and by Reference	131
Chapter 15 Mathematical Functions	135
15.1 The Abs Function	135
Example 15.1	135
15.2 The Exp function	136
Example 15.2	136
15.3 The Fix Function	137
Example 15.3	138
15.4 The Int Function	138
15.5 The Log Function	129
Example 15.4	139
15.6 The Rnd() Function	140
Example 15.5	140
15.7 The Round Function	141
Example 15.6	141
Chapter 16 The Format Function	143

16.1 Format Function for Numbers	143
16.1(a) Built-in Format function for Numbers	143
Example 16.1	144
16.1(b) User-Defined Format	145
Example 16.2	147
16.2 Formatting Date and Time	148
16.2(a) Formatting Date and time using predefined formats	148
Example 16.3	149
16.2(b) Formatting Date and time using user-defined formats	150
Example 16.4	151
Chapter 17 Using Checkbox and Radio Button	153
17.1 Check Box	153
Example 17.1: Shopping Cart	153
Example 17.2	155
Example 17.3	156
17.2 Radio Button	158
Example 17.4	158
Example 17.5	160
Chapter 18 Errors Handling	163
18.1 Introduction to Object Oriented Programming	163
18.2 Using On Error GoTo Syntax	164
Example 18.1: Division Errors	164
18.3 Errors Handling using Try...Catch ...End Try Structure	165
Chapter 19 Object Oriented Programming	168
19.1 Concepts of Object Oriented Programming	168
19.2 Creating Class	169
Chapter 20 Creating Graphics	174
20.1 Introduction to Graphics Creation	174
20.2 Creating the Graphics Object	175

20.3 Creating the Pen Object	176
20.4 Drawing a Line	176
20.5 Drawing Lines that Connect Multiple Points	178
Example 20.1	178
20.6 Drawing a curve that Connect Multiple Points	179
Example 20.2	180
20.7 Drawing a Quadratic Curve	181
Example 20.3	181
20.8 Drawing a Sine Curve	183
Example 20.4	184
20.8 Drawing a Rectangle	185
20.6 Customizing Line Style of the Pen Object	187
Example 20.5	187
20.7 Drawing an Ellipse	188
Example 20.6	190
Example 20.7	191
20.8 Drawing a Circle	191
Example 20.8	191
Example 20.9	192
20.9 Drawing Text	192
Example 20.10	194
Example 20.11	195
20.10 Drawing Polygons	196
Example 20.12 Drawing a Triangle	197
Example 20.13 Drawing a Quadrilateral	198
20.11 Drawing a Pie	199
Example 20.14 Drawing a pie that sweeps clockwise through 60 degrees.	200
20.12 Filling Shapes with Color	201
20.12(a) Drawing and Filling a Rectangle with Color	201

Example 20.15	201
20.12(b) Drawing and Filling an Ellipse with Color	203
Example 20.16	203
20.12(c) Drawing and Filling a Polygon with Color	204
Example 20.16	204
20.12(d) Drawing and Filling a Pie	205
Example 20.18	206
Chapter 21 Using Timer	208
21.1 Creating a Digital Clock	208
21.2 Creating a Stopwatch	209
21.3 Creating a Digital Dice	211
Chapter 22 Creating Animation	214
22.1 Creating Motion	214
22.2 Creating a Graphical Dice	216
22.3 Creating a Slot Machine	218
Chapter 23 Working with Databases	222
23.1 Introduction to Database	222
23.2 Creating a Database Application	223
23.3 Creating a Connection to a Database using ADO.NET	225
23.4 Populating Data in ADO.NET	232
Example 23.1	234
23.5 Browsing Records	236
23.6 Editing, Saving, Adding and Deleting Records	237
Example 23.2	238
23.7 Accessing Database using DataGridView	241
Example 23.3	242
23.8 Performing Arithmetic Calculations in a Database	243
Example 23.4	243
Example 23.5	245

Example 23.6	247
Chapter 24 Reading and Writing Text Files	249
24.1 Introduction	249
24.2 Reading a Text File	249
24.3 Writing to a Text File	254
Chapter 25 Building Console Applications	257
25.1 Introduction	257
Example 25.1: Displaying a Message	260
25.2 Creating a Text File Writer in Console	261
Example 25.2	261
25.3 Creating a Text File Reader in Console	262
Example 25.3	262
25.4 Creating a Console App using If...Then....Else	263
Example 25.4	263
Chapter 26 Creating Menu Bar and ToolBar	265
26.1 Creating Menu Items on the Menu Bar	265
26.2 Creating The Toolbar	275
27. Deploying Your VB2019 Applications	281

Chapter 1

Introduction to Visual Basic 2019

- ❖ A brief description of Visual Basic 2019
- ❖ Getting to know the Visual Basic 2019 Integrated Development Environment

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. Therefore, 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 VB.NET programming language released by Microsoft.

Learn more about Visual Studio 2019 from the Youtube link below:

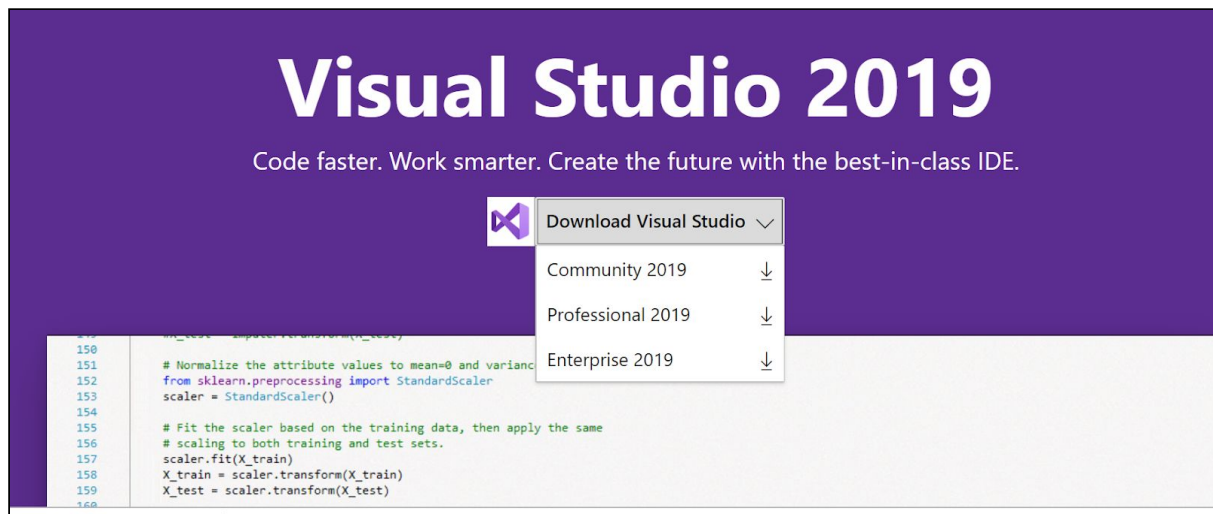
<https://youtu.be/n5sJ4EewKGk>

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 below:



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 that provides full-featured IDE for students, open source community and individuals. As 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 1.1

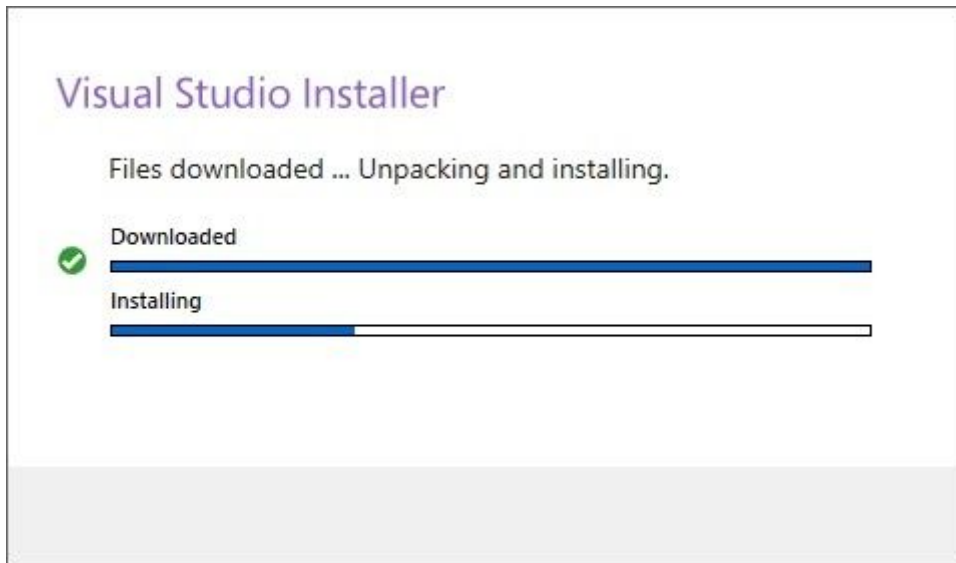


Figure 1.1

You'll see several status screens that show the progress of the installation. After the installer has finished installing, it's time to pick the feature set that you wish to install, as shown in Figure 1.2. 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.

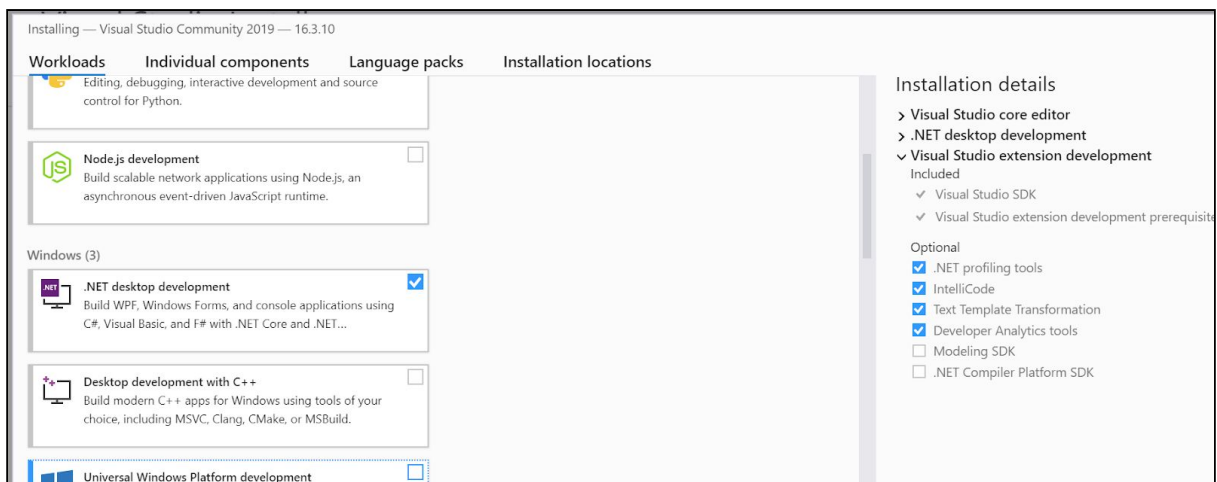


Figure 1.2

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

Launching Microsoft Visual Studio 2019 will bring you to the Visual Studio 2019 Start Page, as shown in Figure 1.3

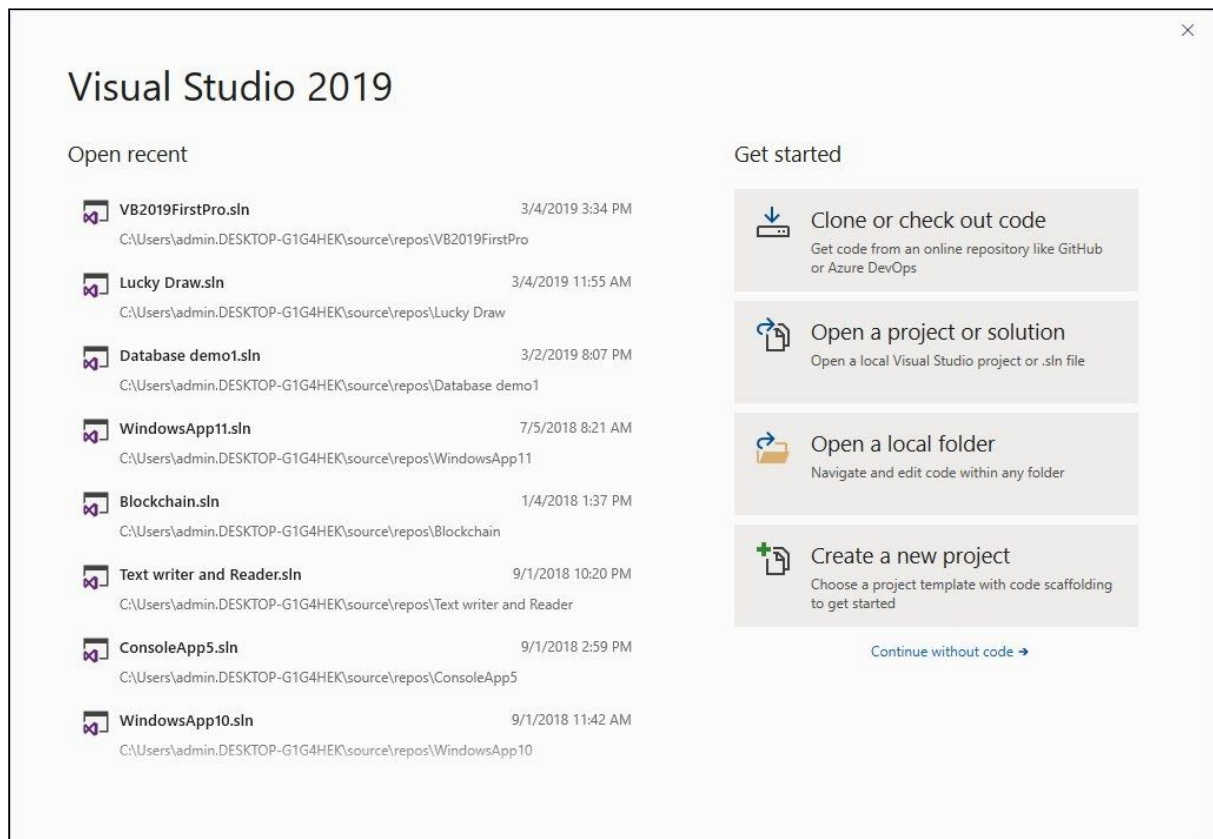


Figure 1.3 Visual Studio 2019 Start Page

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's create a new project by clicking on the Create a new project option. You will now see the Create a new project template page, as shown in Figure 1.4. In the Create a new project page, select the Visual Basic language.

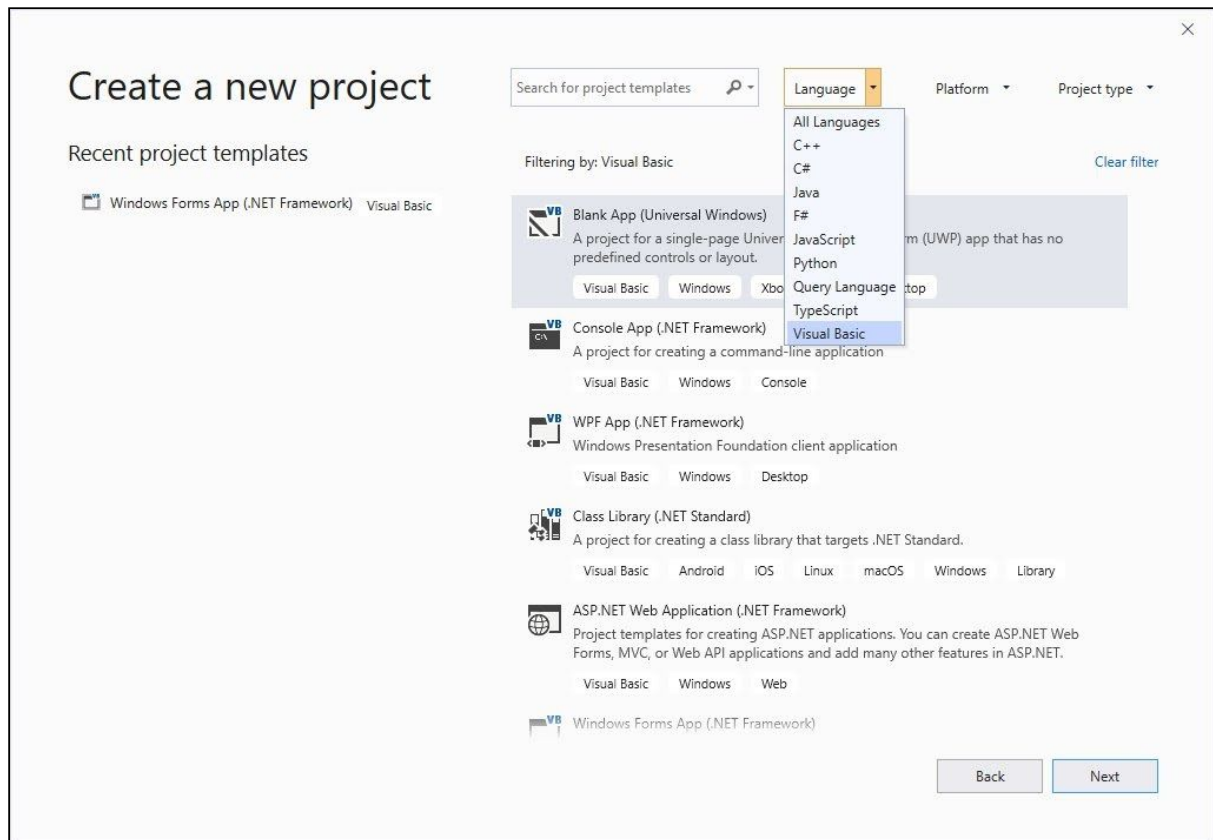


Figure 1.4 Create a new project template

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

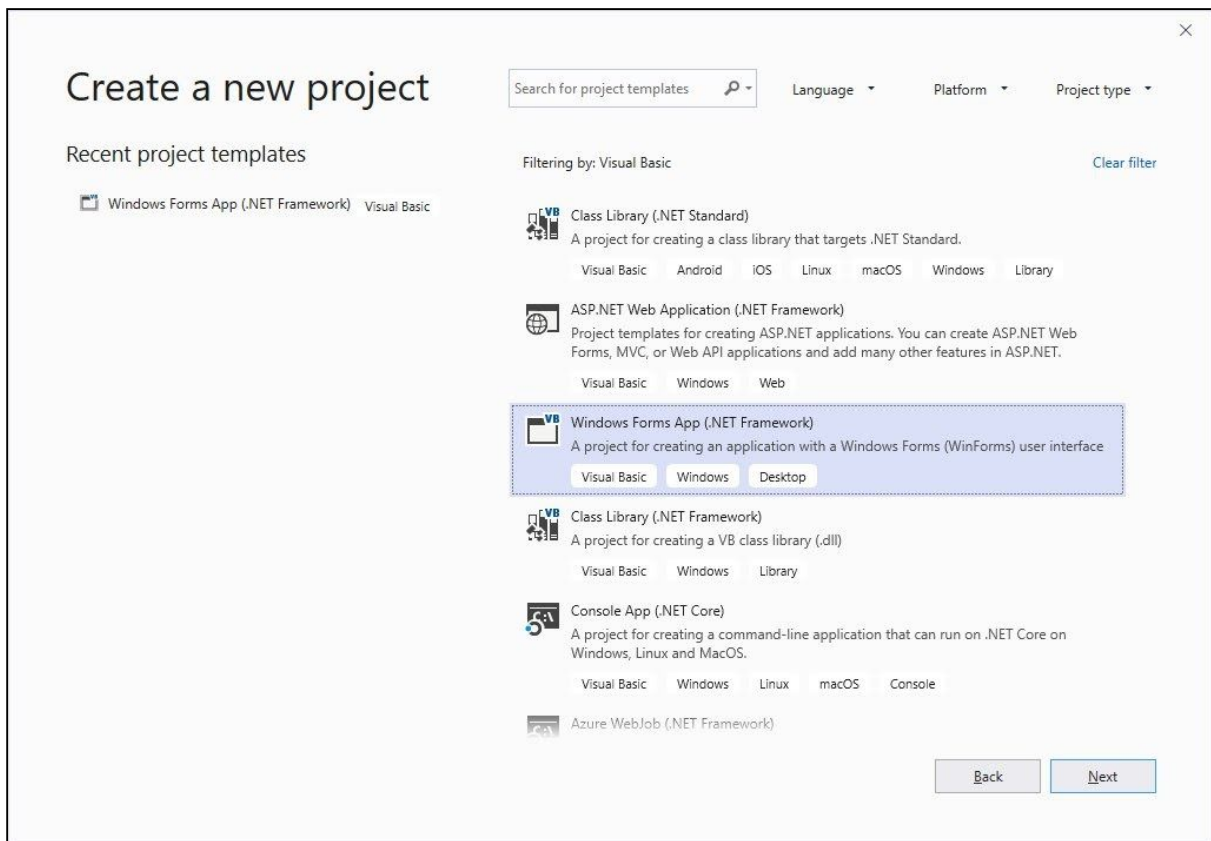


Figure 1.5 Create a new project template

Upon clicking the selected project template, the project configuration page appears, as shown in Figure 1.6. 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) Visual Basic Windows Desktop

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 1.6 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 1.7. 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.

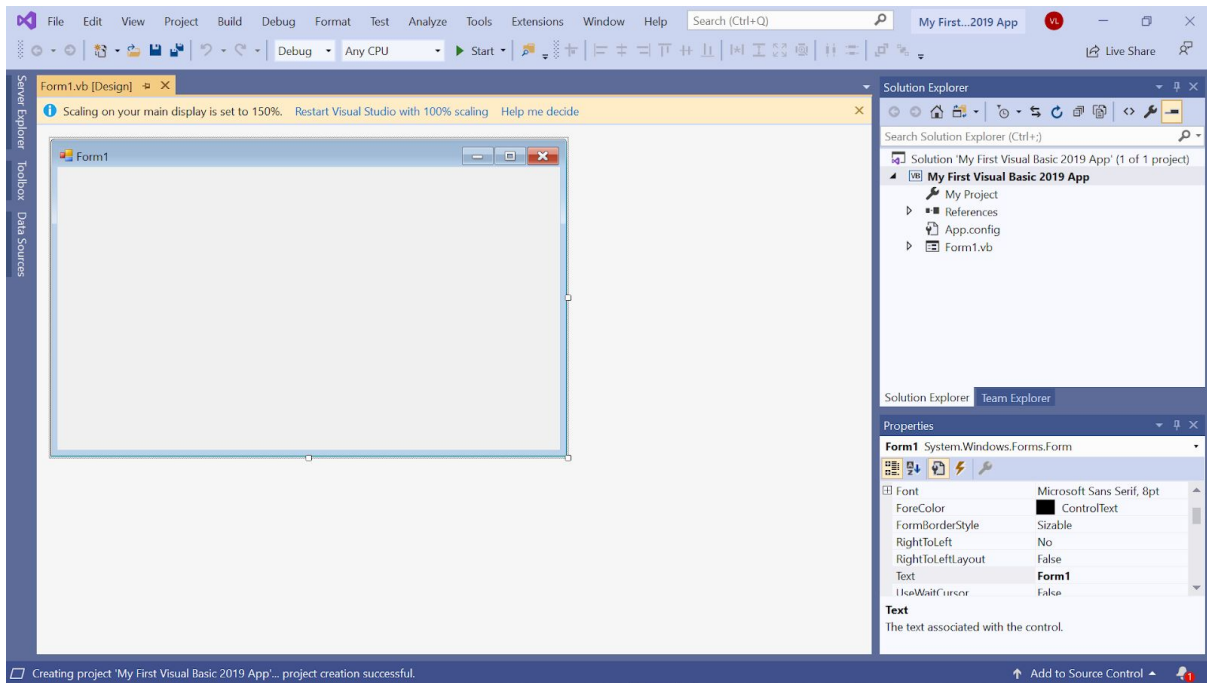


Figure 1.7 The Visual Basic 2019 Express IDE

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 1.8. You can drag and move your toolbox around and dock it to the right, left, top or bottom of the IDE.

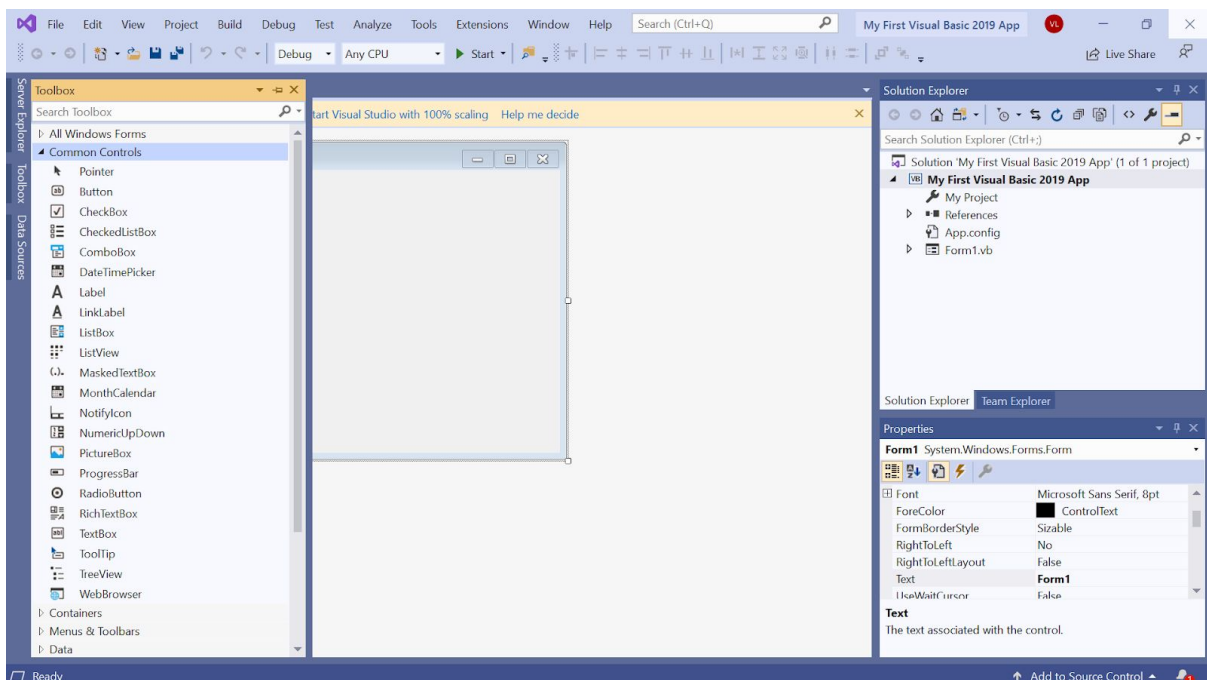


Figure 1.8 Visual Basic 2019 Tool Box

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 1.9

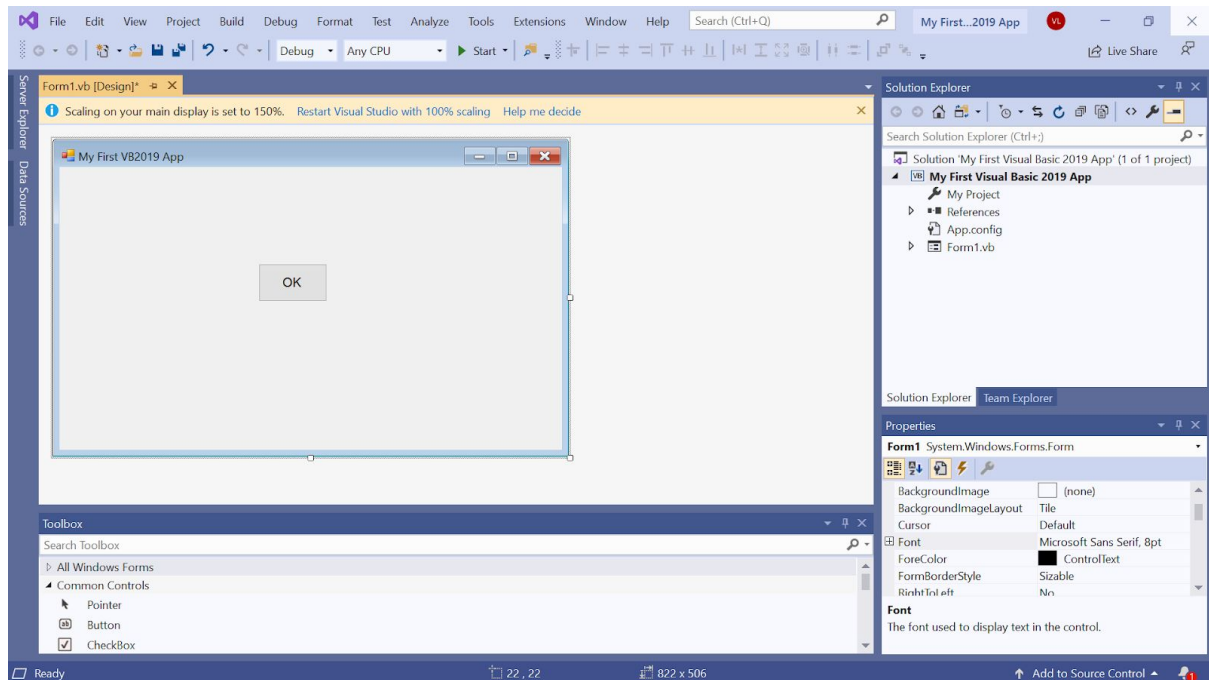


Figure 1.9 The Design Interface

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 1.10.

```
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 1.11. 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 1.12. The function **MsgBox** is a built-in function of Visual Basic 2019 which can display the text enclosed within the brackets.

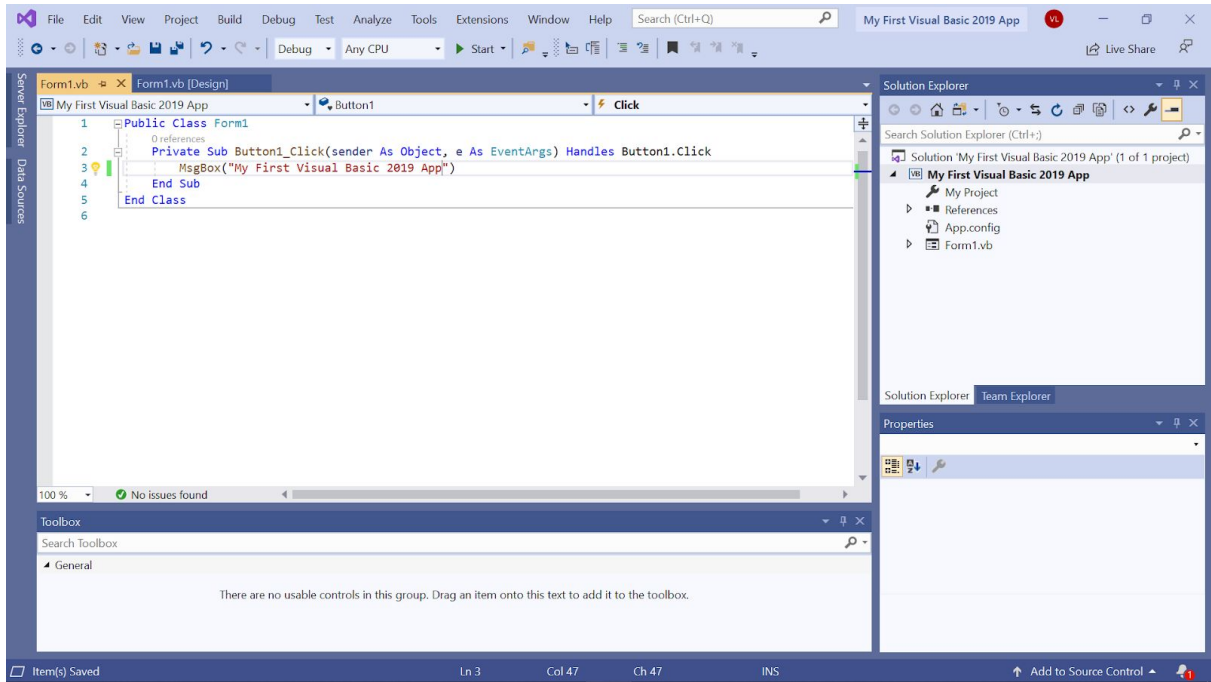


Figure 1.10 Visual Basic 2019 Code Window

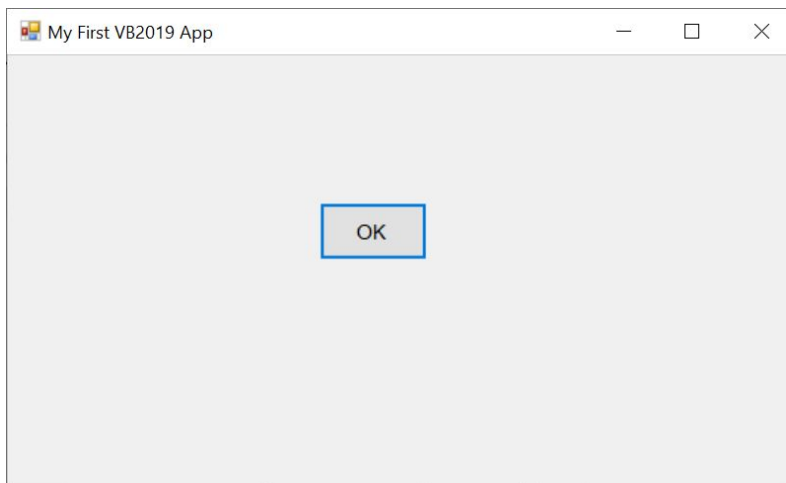


Figure 1.11 The Runtime Interface

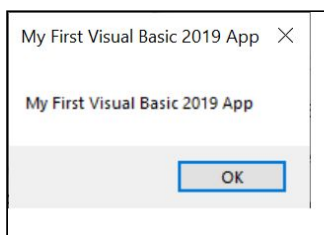


Figure 1.12 The Message Box

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

- ❖ Customizing
- ❖ Adding controls
- ❖ Setting Control Properties

As Visual Basic 2019 is a GUI-based programming language, the first step in developing an application is to design the user interface(UI). To build a graphical user interface, first of all 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 related to 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 following properties:

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

In fact, 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. For example, an RGB value of 128, 255, 255 yield the cyan color.

On the other hand, 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 (in hexadecimal notation), 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 of VS2019, it automatically converts the color to RGB color or the color name. 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 2.2 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 LavenderBlush , the text OK color is Crimson and the window cannot be maximized.

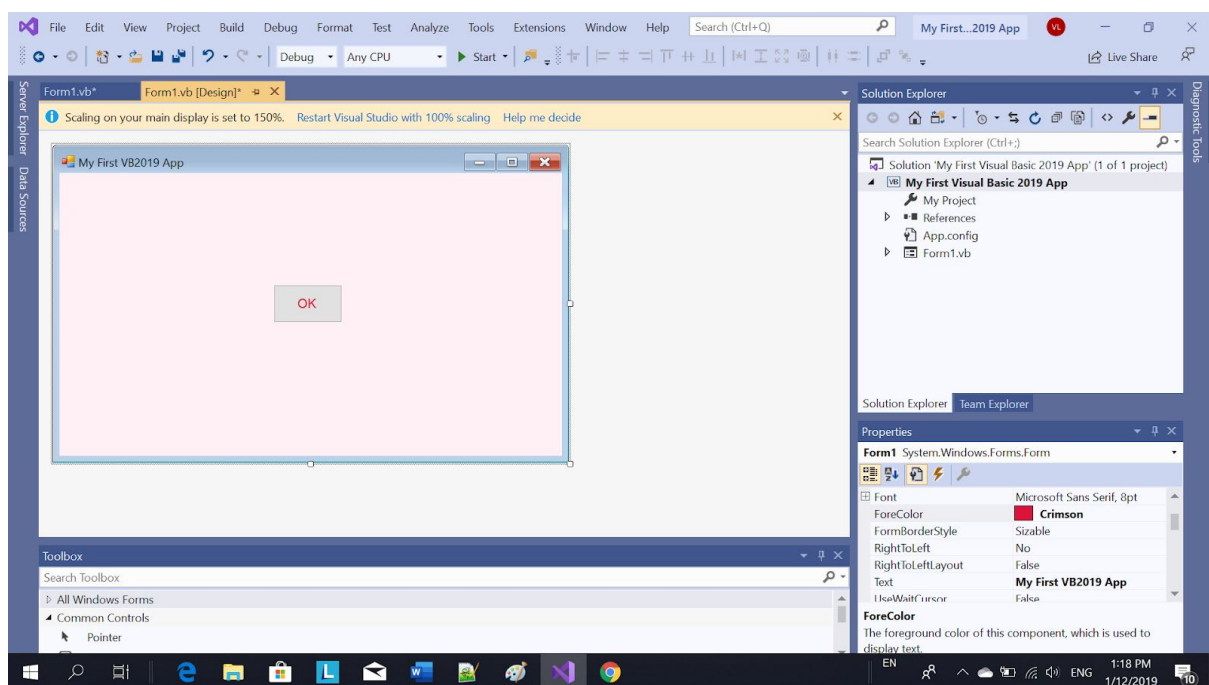


Figure 2.3 Design UI

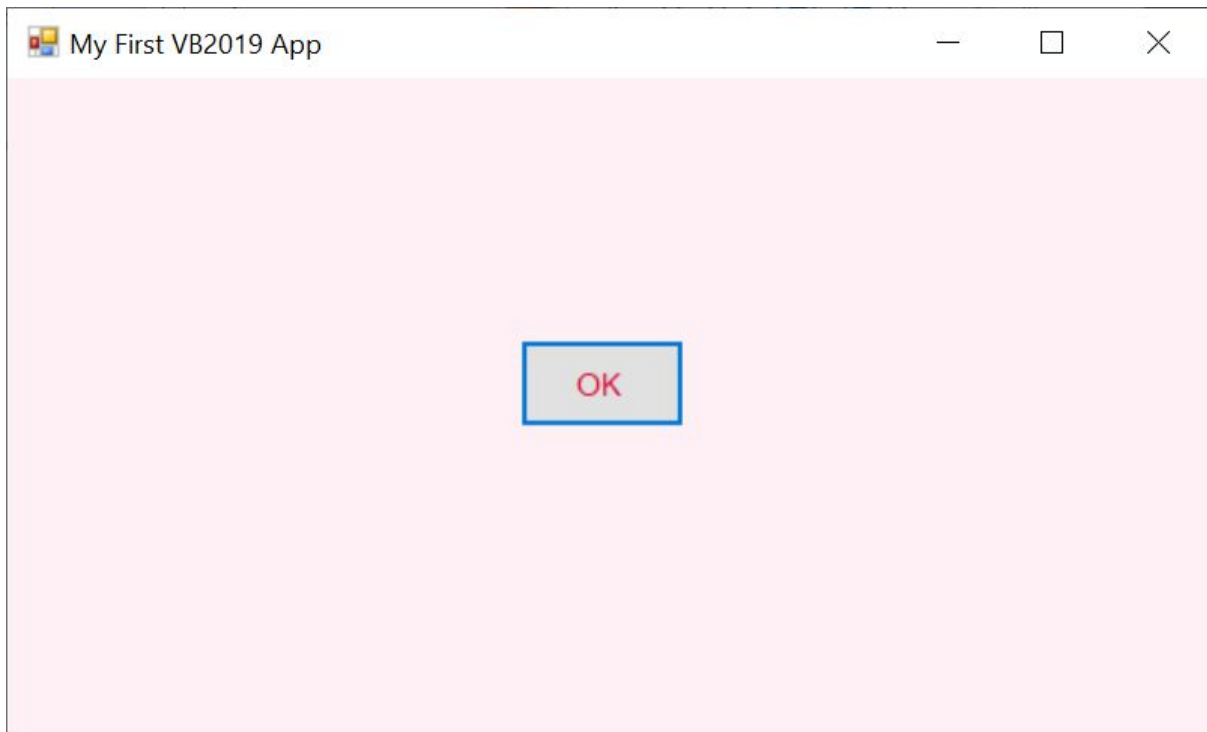


Figure 2.4 Runtime UI

You can also change the properties of the form at runtime by writing the 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 dot or period:

`ObjectName.property`

For example, we can set the background of the form to blue using the following code:

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

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

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

Now, type in the following code by clicking the form to enter the code window:

```
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 is now different from that shown in Figure 2.4,

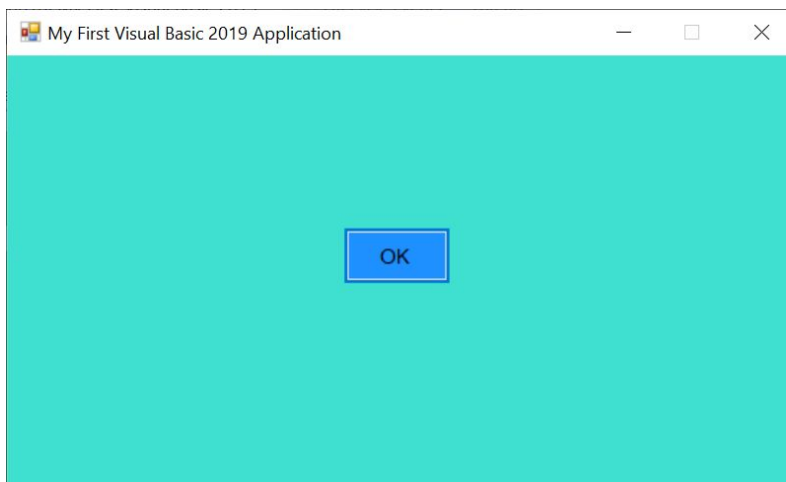


Figure 2.5

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

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

In addition, you can also specify the size, the opacity and the position of the default form using the code, as follows:

```
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

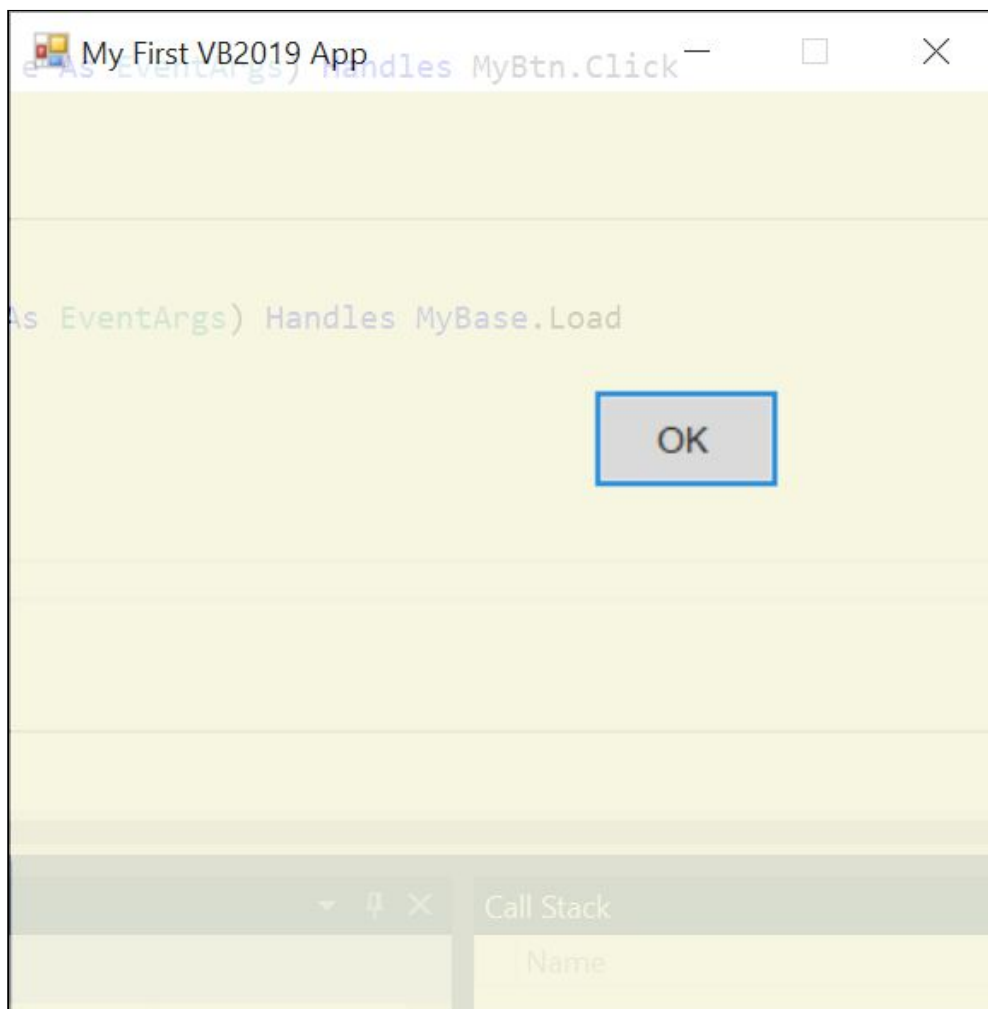


Figure 2.6

2.2 Adding Controls to the Form

In section 2.1, we have learned how to build an initial UI in Visual Basic 2019 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. They can be added to the form from the Toolbox. Among the controls, the most common ones are the button, label, textbox, listbox, combobox, picture box, checkbox, 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 tool box, as shown in Figure 2.6. You can also use shortcut keys Ctrl+w+x to bring out the toolbox.

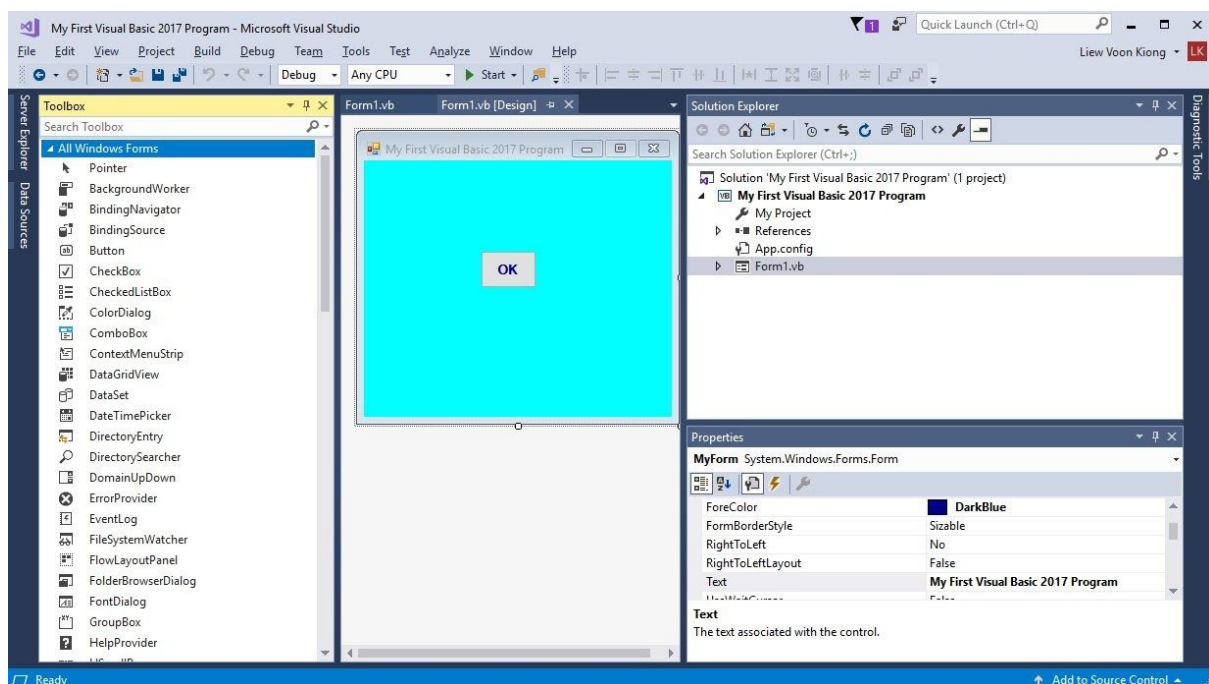


Figure 2.6: Toolbox

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 tool box and choose dock from the pop-up menu. The docked Toolbox that appears side by side with the Solution Explorer, and as one of the tabbed windows together with the Form Design window and the code window, as shown in Figure 2.7.

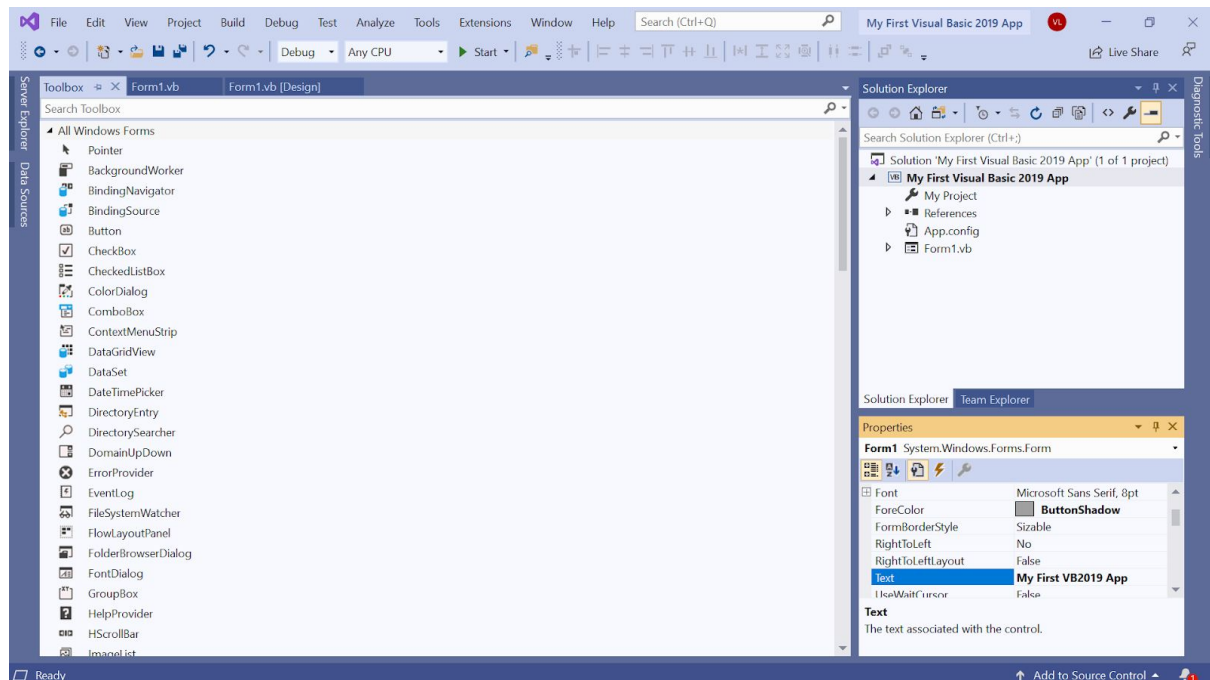


Figure 2.7 Toolbox

You can also dock the tool box at the bottom, below the default form, as shown in Figure 2.8. Further, you may also pin the tool box 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 tool box is entirely up to you but we strongly suggest that you place the tool box alongside or at the bottom of the default form so that it is easy for you to add controls from the tool box into the form. You should never cover the form with the toolbox because it will be difficult to add controls to the form.

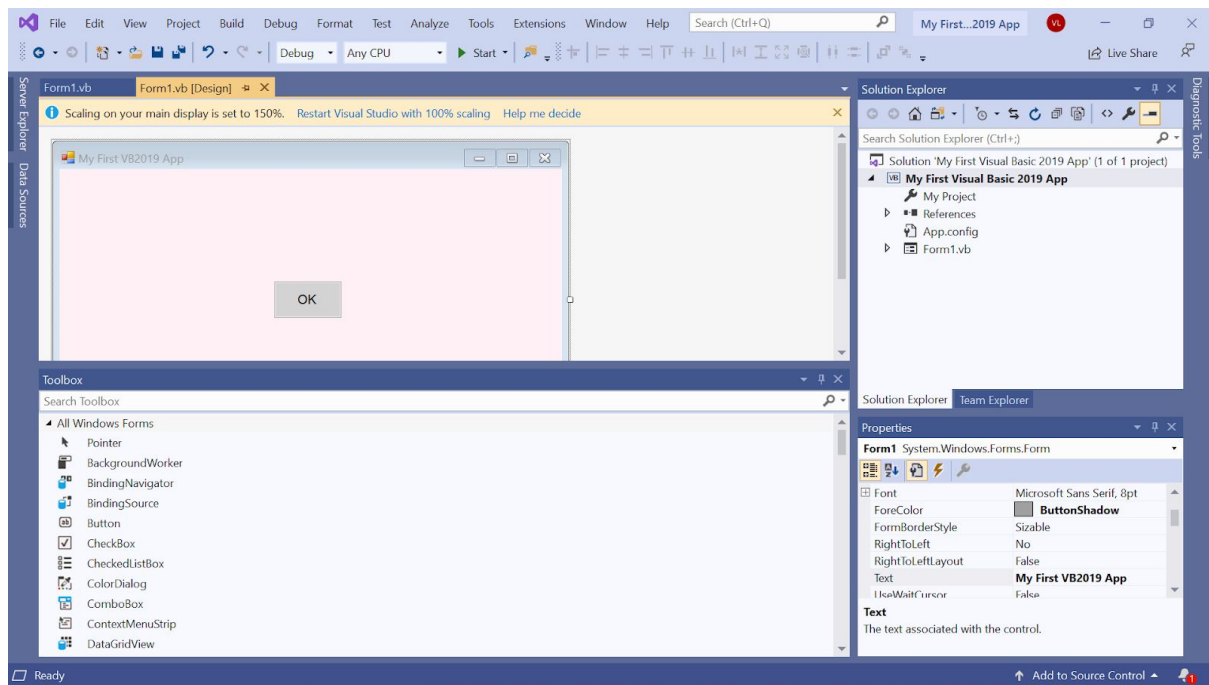


Figure 2.8

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 you can also 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.

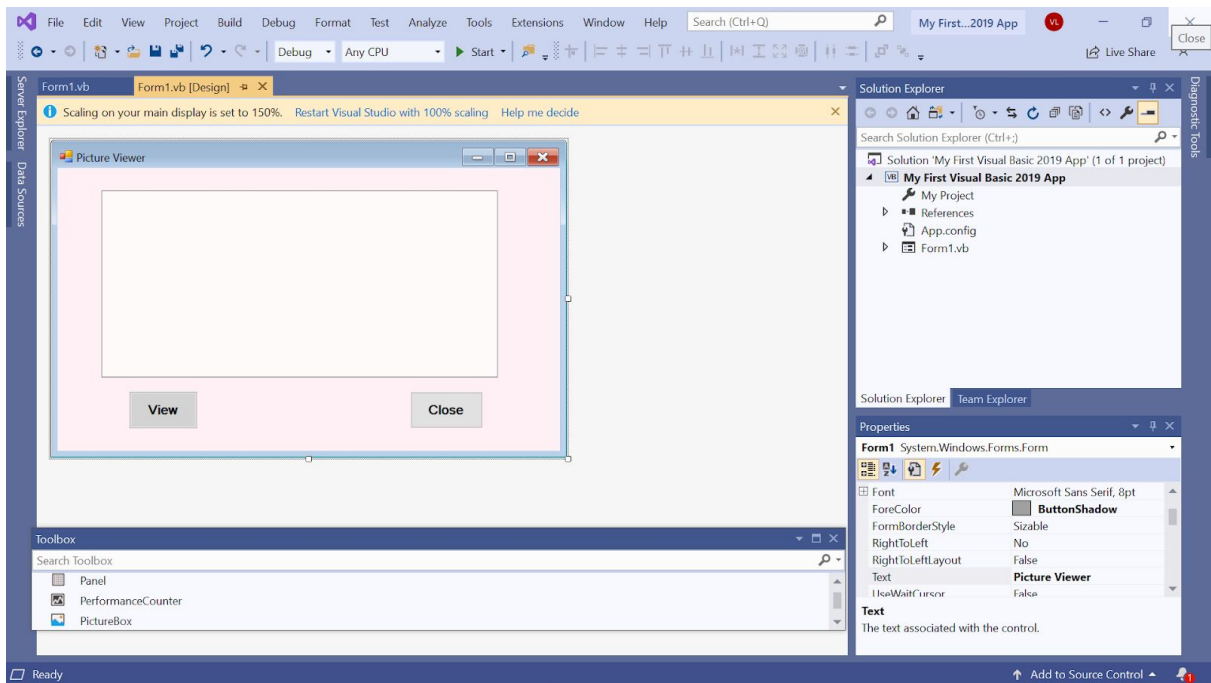


Figure 2.9

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.