Hello User Formative Assessment

Objective: To determine your ability to follow directions and execute basic code. No knowledge necessary…just do the work and follow the directions carefully! Pay attention to detail.

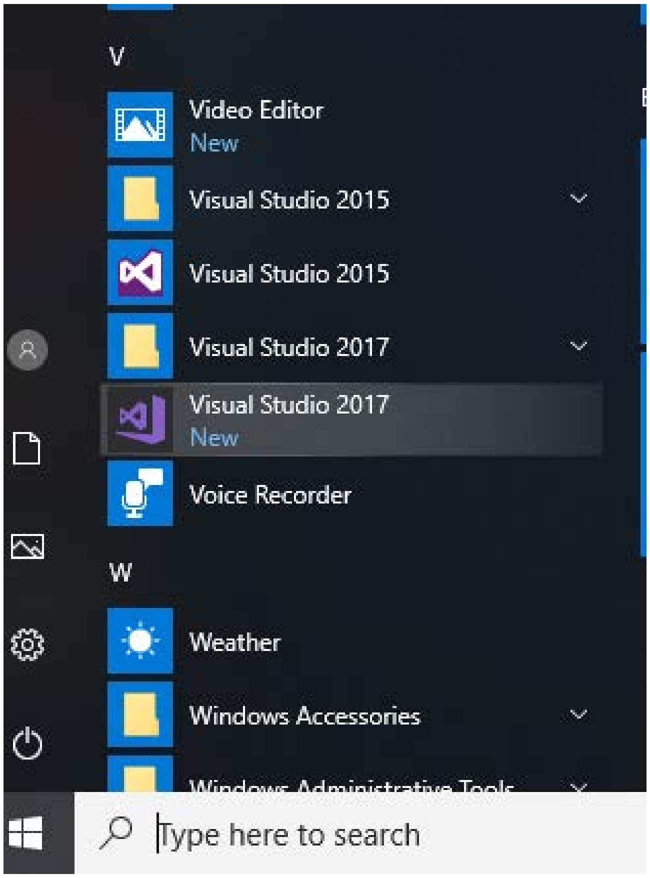
A formative assessment allows a teacher to see exactly where you are with a particular skill before you are expected to know the skill—in this case, we're measuring your basic aptitude with computers and ability to interpret written directions and read/view ALL the steps. There are checkpoints at various places where you need to report your progress to Mr. Fox. These will be noted in **red, bold text**. Raise your hand when you are ready to be checked at each point.

You're going to create your first Visual Basic program today (take as much of the period as you need). Like all parts of this course, it is very important that you follow instructions exactly as they appear.

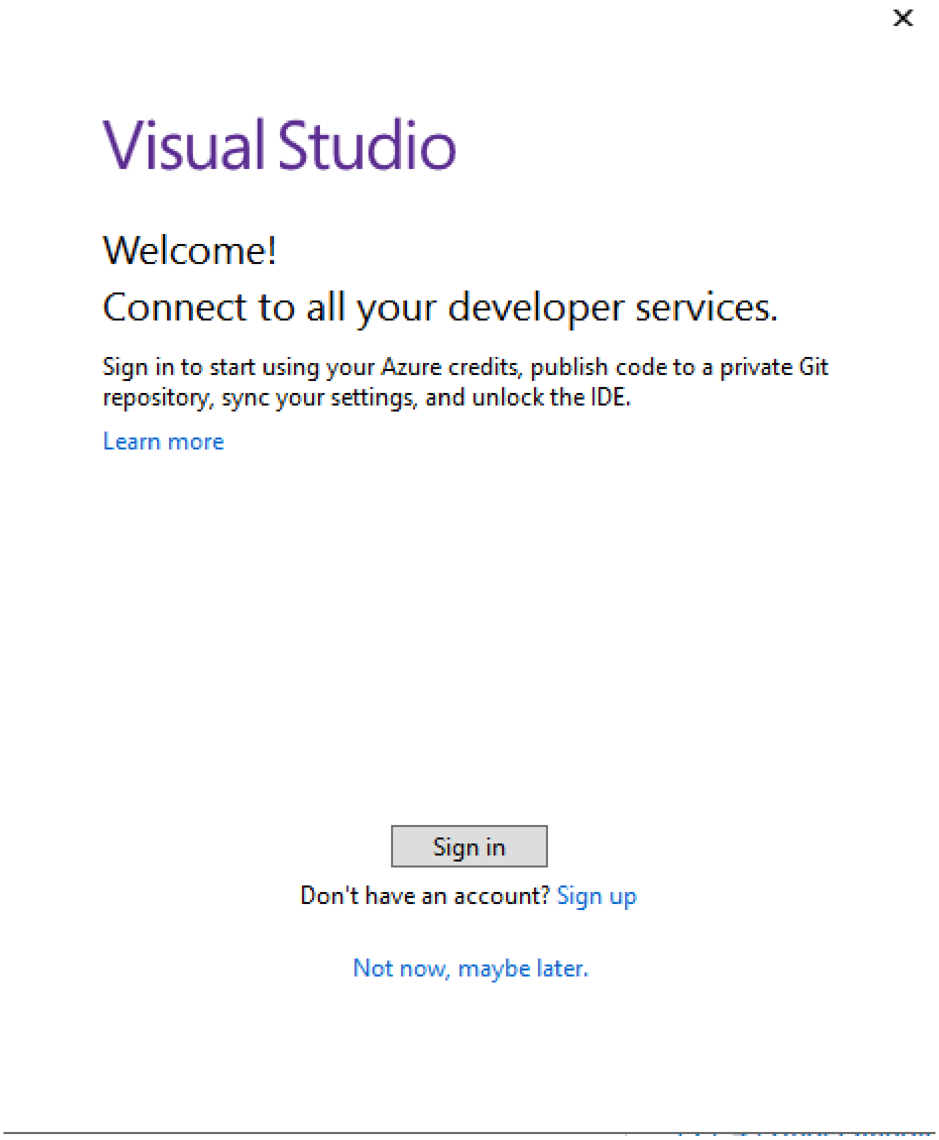
Directions

You are not allowed to receive help from other students for this assessment.

1. Login to the network.
2. If you have problems logging in, please let Mr. Fox know if you're having problems logging in.
3. Once you've logged in: Locate the **Visual Studio 2017** program (note, do not locate **Blend for Visual Studio**). For Windows 10, this includes the ability to pull up the Start Menu (as noted below) or to search for Visual Studio 2017.



4. Once you have started Visual Studio 2017, the following screen will pop up (do not sign in); click on **`Not Now Maybe Later`**. If you don’t get this screen, it simply means that someone in an earlier class may have already configured Visual Studio 2017.



Visual Studio

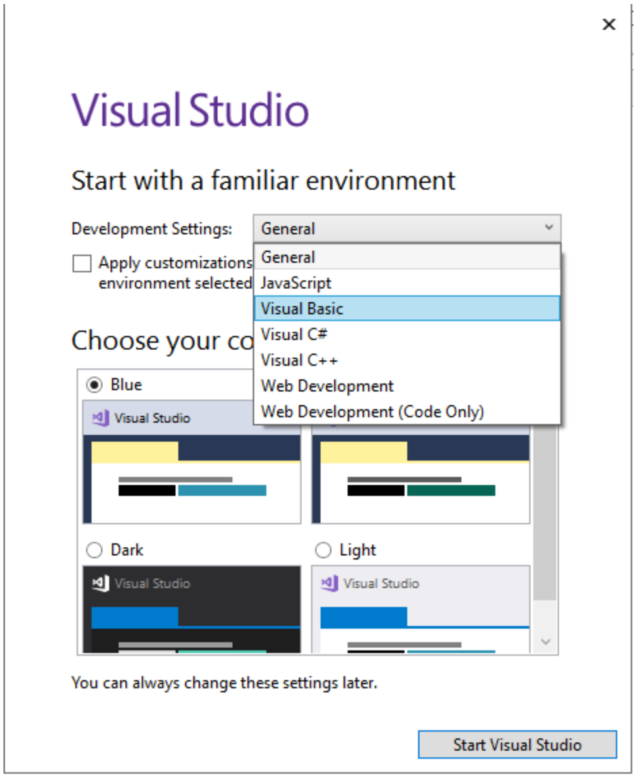
Welcome!

Connect to all your developer services.

Sign in to start using your Azure credits, publish code to a private Git repository, sync your settings, and unlock the IDE.

Don't have an account? Sign up

5. Choose to install “Visual Basic” from the drop down menu next to **`Development Settings`** (see the figure below).



Visual Studio

Start with a familiar environment

Development Settings: General

□ Apply customizations environment selected

Choose your co

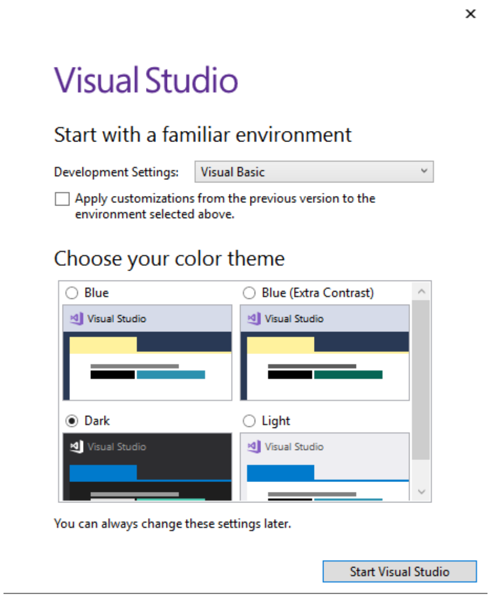
● Blue

○ Dark

○ Light

You can always change these settings later.

6. Choose the “dark” configuration (lower left) since it will look like the examples you will see (and it’s the *dark side* which is always a little more fun and stylish). If you choose another configuration, it will not look the same and you won’t have nearly the same amount of fun as everyone else.



Visual Studio

Start with a familiar environment

Development Settings: Visual Basic

□ Apply customizations from the previous version to the environment selected above.

Choose your color theme

○ Blue

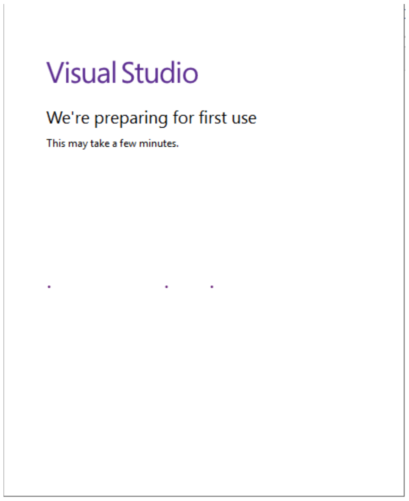
○ Blue (Extra Contrast)

● Dark

○ Light

You can always change these settings later.

7. Click “Start Visual Studio.” You’ll get this screen for a little bit…or a little bit longer. Again, if you’re reading this and you didn’t get this, skip on down to step

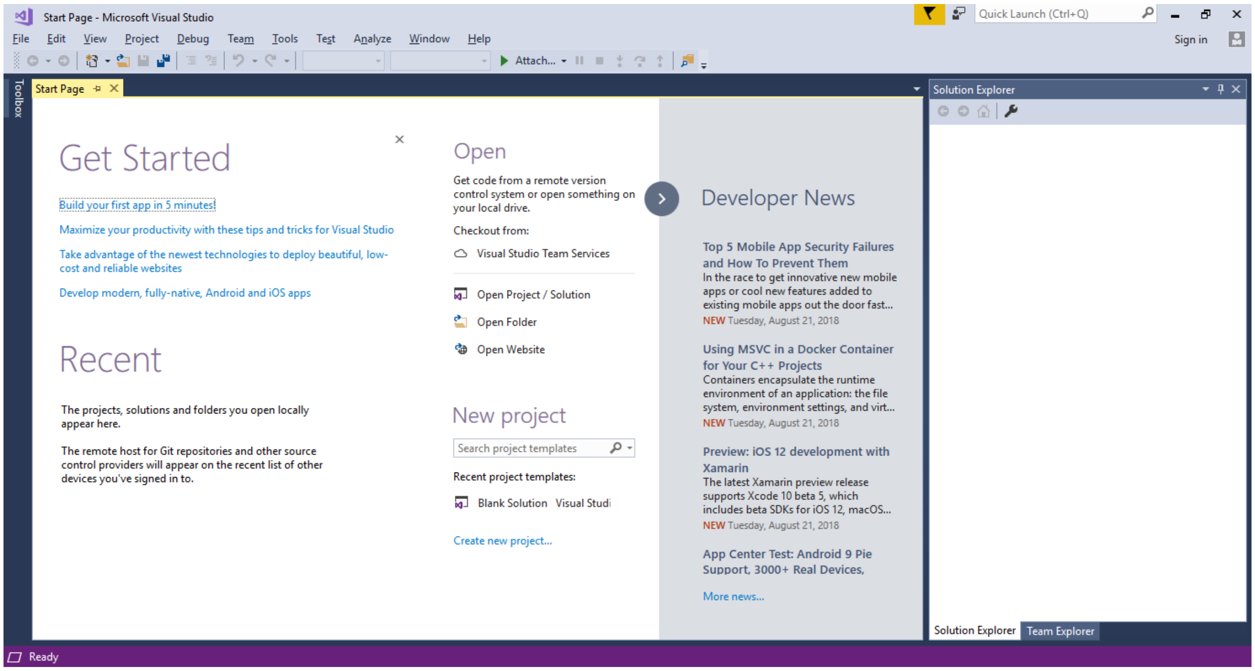


Visual Studio

We're preparing for first use

This may take a few minutes.

8. Then you’ll get this screen. Click the “New Project” button on the toolbar. You can also select this by going to “File → New Project.” There are actually several ways you can create a new project.



Get Started

Build your first app in 5 minutes!

Maximize your productivity with these tips and tricks for Visual Studio

Take advantage of the newest technologies to deploy beautiful, low-cost and reliable websites

Develop modern, fully-native, Android and iOS apps

Recent

The projects, solutions and folders you open locally appear here.

The remote host for Git repositories and other source control providers will appear on the recent list of other devices you've signed in to.

Open

Get code from a remote version control system or open something on your local drive.

Checkout from:

Visual Studio Team Services

Open Project / Solution

Open Folder

Open Website

New project

Search project templates

Recent project templates:

Blank Solution Visual Studi

Create new project...

Developer News

Top 5 Mobile App Security Failures and How To Prevent Them
In the race to get innovative new mobile apps or cool new features added to existing mobile apps out the door fast...
NEW Tuesday, August 21, 2018

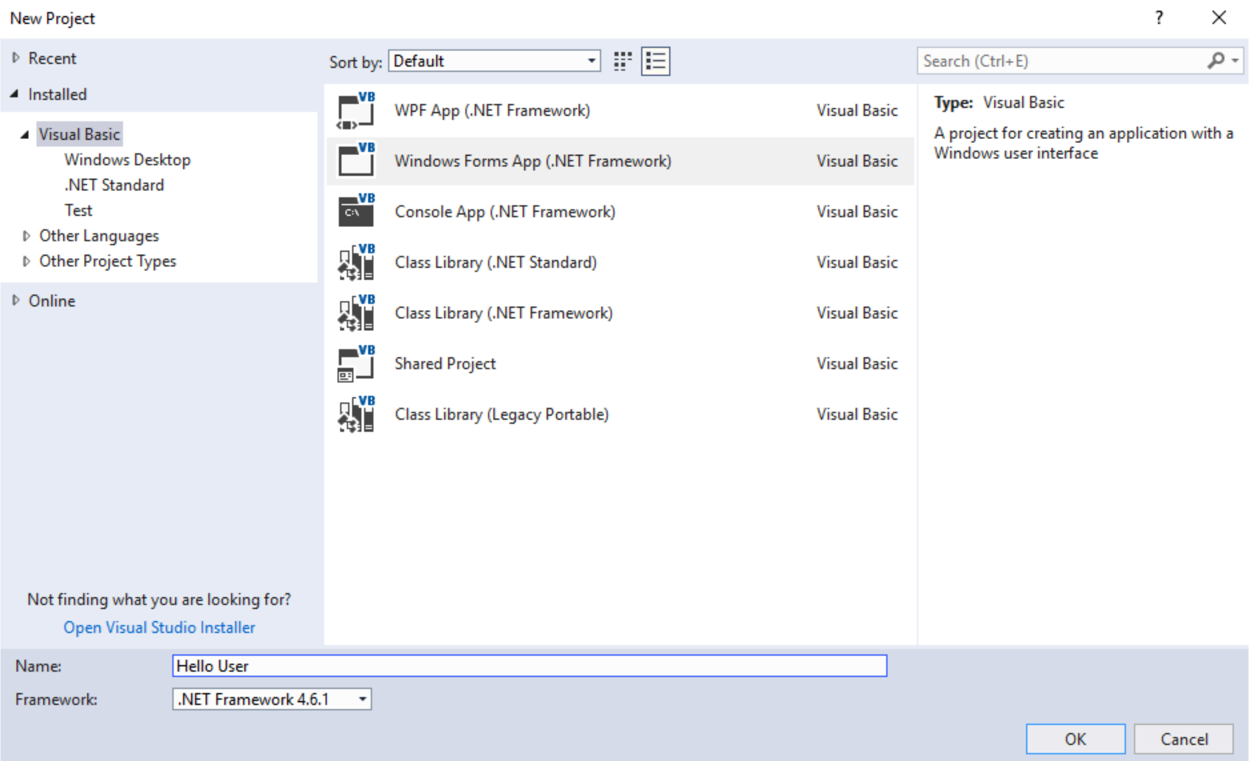
Using MSVC in a Docker Container for Your C++ Projects
Containers encapsulate the runtime environment of an application: the file system, environment settings, and virt...
NEW Tuesday, August 21, 2018

Preview: iOS 12 development with Xamarin
The latest Xamarin preview release supports Xcode 10 beta 5, which includes beta SDKs for iOS 12, macOS...
NEW Tuesday, August 21, 2018

App Center Test: Android 9 Pie Support, 3000+ Real Devices,

More news...

9. In the New Project dialog, select **`Visual Basic`** in the Installed Template tree-view box to the left. The Templates pane on the right will display all the available templates for the project type chosen. Select the **`Windows Forms App (.NET Framework)`** template. You can type **`Hello User`** in the name box at the bottom. After that click **OK**.



New Project

Sort by: Default

WPF App (.NET Framework)	Visual Basic
Windows Forms App (.NET Framework)	Visual Basic
Console App (.NET Framework)	Visual Basic
Class Library (.NET Standard)	Visual Basic
Class Library (.NET Framework)	Visual Basic
Shared Project	Visual Basic
Class Library (Legacy Portable)	Visual Basic

Type: Visual Basic

A project for creating an application with a Windows user interface

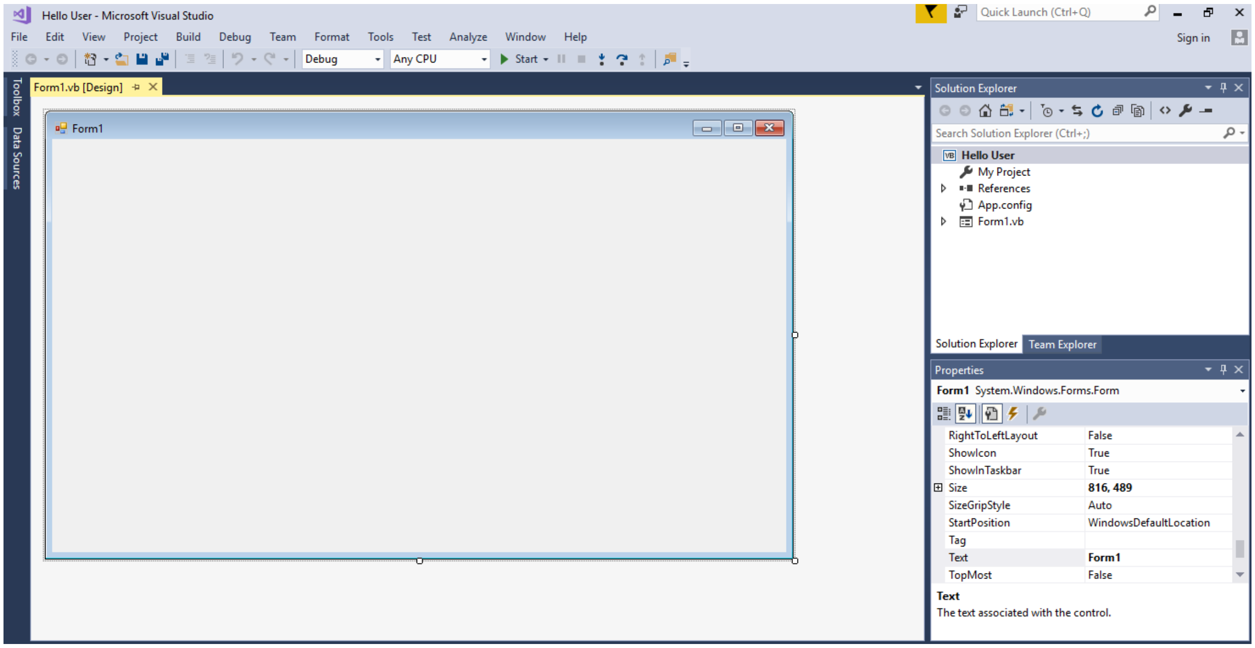
Not finding what you are looking for?

Open Visual Studio Installer

Name: Hello User

Framework: .NET Framework 4.6.1

10. Now, the IDE creates an empty Windows application for you. So far, your Hello User program consists of one blank window, called a Windows Form (or sometimes just a form), with the default name of “Form1.vb” like so:



Form1 System.Windows.Forms.Form

RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	816, 489
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	Form1
TopMost	False

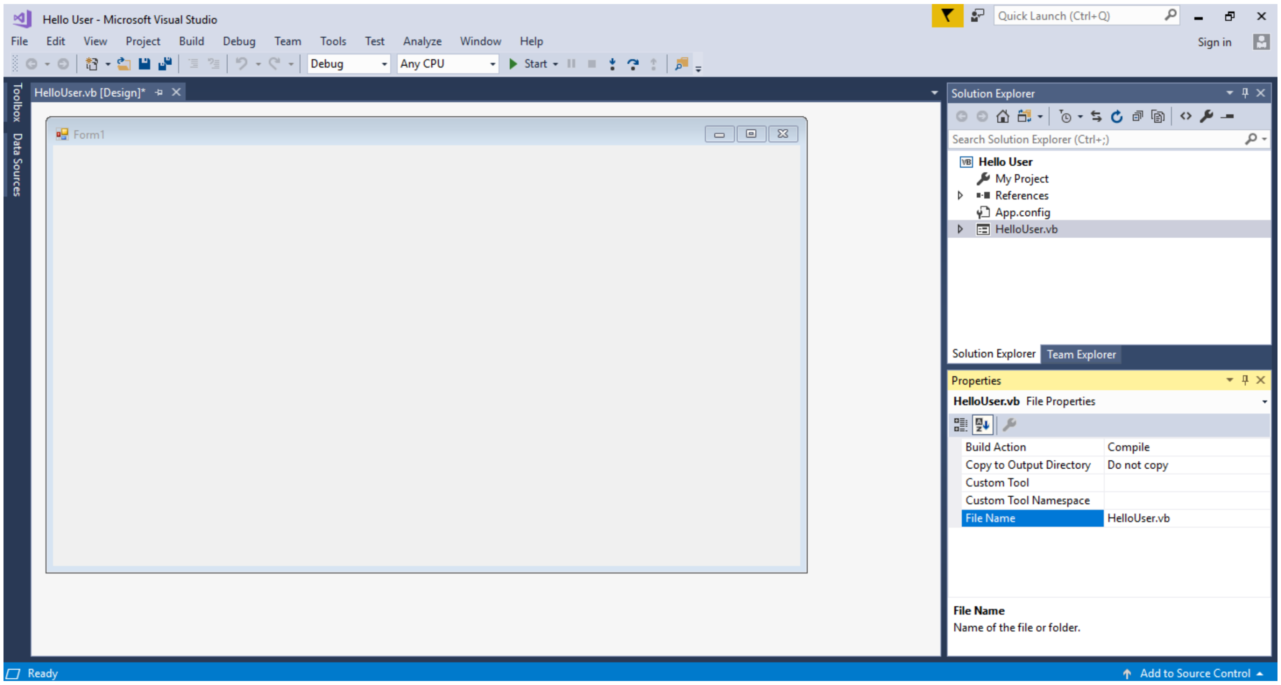
Text
The text associated with the control.

CHECK #1: CALL MR. FOX OVER AND SHOW HIM THAT YOU’VE REACHED THIS POINT (YOUR SCREEN SHOULD LOOK LIKE THE SCREEN ABOVE).

Once Mr. Fox has checked you off, proceed to the next page.

11. Next, you’ll give your form a name and set a few properties for it:

A. Change the name of your form to something more indicative of your application. Click **`Form1.vb`** in the Solution Explorer window. Then, in the **`Properties`** window, change the **`File Name`** property from **`Form1.vb`** to **`HelloUser.vb`** and press Enter (the properties will not take effect if you don’t press enter or click another property).

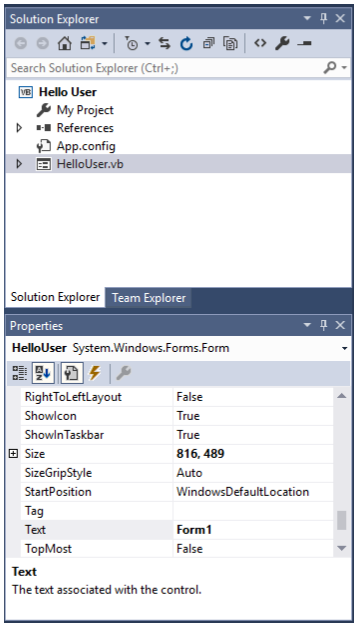


HelloUser.vb File Properties

Build Action	Compile
Copy to Output Directory	Do not copy
Custom Tool	
Custom Tool Namespace	
File Name	HelloUser.vb

File Name
Name of the file or folder.

B. Click the form displayed in the Design window. The Properties window changes to display the form’s Form properties.



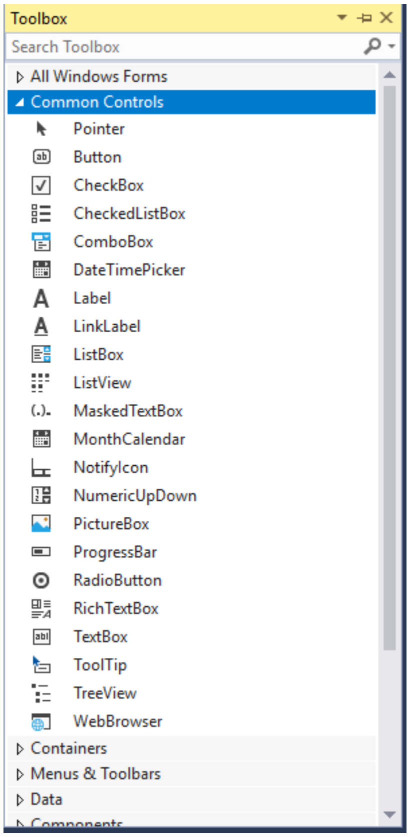
HelloUser System.Windows.Forms.Form

RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
⊞ Size	816, 489
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	Form1
TopMost	False

Text
The text associated with the control.

C. Right now, the title (**Text** property) of your form (displayed in the bar at the top of the form) is **Form1**. This is not very descriptive, so change it to reflect the purpose of this application. Locate the text property in the Properties window. Change the Text property's value to **Hello from Visual Basic 2017** and press enter. Note that the form's title has been updated to reflect the change.

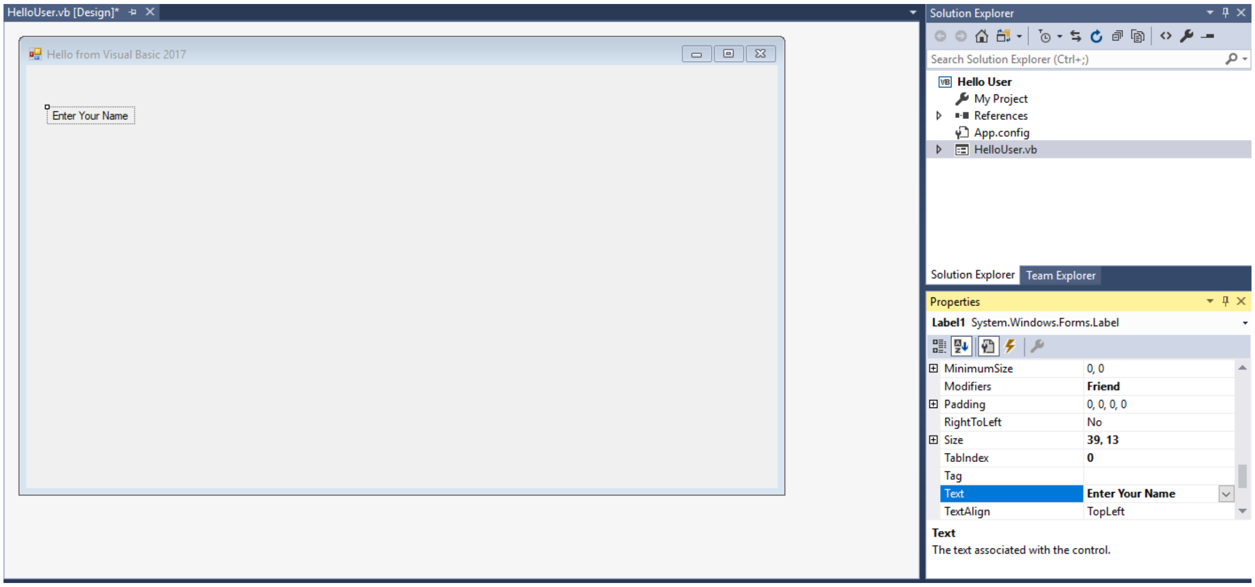
12. Next, open the toolbox. You do this through the "View → Toolbox" menu option, by clicking the Toolbox icon on the Standard menu bar, or by pressing Ctrl + Alt + X. Also, the Toolbox tab appears on the left of the IDE; hovering your mouse over this tab causes the Toolbox window to fly out, partially covering your form.



13. Add a Label control to the form. Click Label in the Toolbox, drag it over to the form's Designer, and drop it in the desired location. When the control is on the form, you can resize it or move it around. To move it, you can click the control and drag it to the desired location.

14. After drawing a control on the form, you should configure its name and the text it will display. You will see that the Properties window to the right of the Designer has changed to **`Label1`**, telling you that you are currently examining the properties for the label. In the Properties window, set your new label's **`Text`** property to **`Enter Your Name`**. Then, set the name property to **`lblName`**.

These are not ones (1) on "lblName", they are lowercase L's.

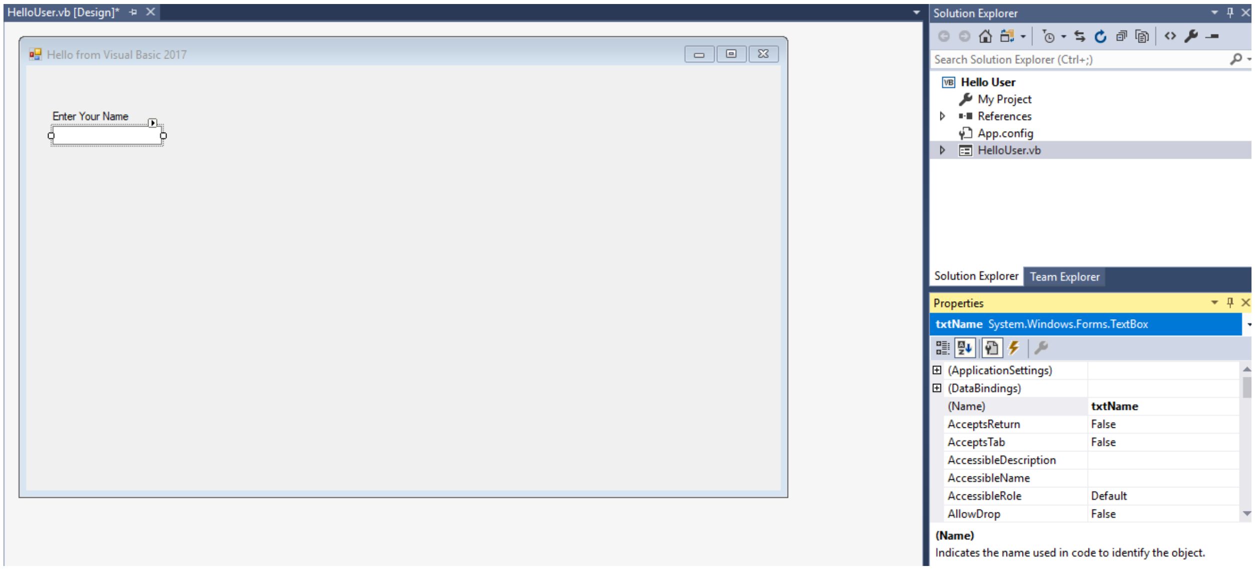


Label1 System.Windows.Forms.Label

⊞ MinimumSize	0, 0
Modifiers	Friend
⊞ Padding	0, 0, 0, 0
RightToLeft	No
⊞ Size	39, 13
TabIndex	0
Tag	
Text	Enter Your Name
TextAlign	TopLeft

Text
The text associated with the control.

15. Directly beneath the label, you want to add a text box so that you can enter a name. You will repeat the procedure you followed for adding the label, but this time make sure you select the **`TextBox`** control from the toolbar. After you have placed the control in the appropriate position, use the Properties window to set its Name property to **`txtName`**.

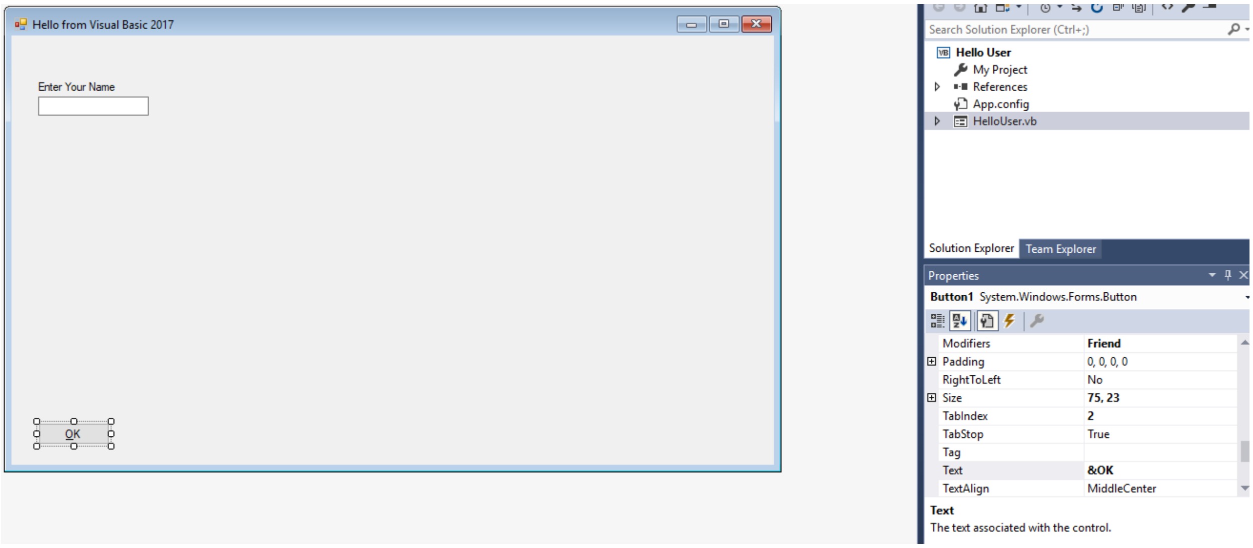


txtName System.Windows.Forms.TextBox

⊞ (ApplicationSettings)	
⊞ (DataBindings)	
(Name)	txtName
AcceptsReturn	False
AcceptsTab	False
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False

(Name)
Indicates the name used in code to identify the object.

16. In the bottom left corner of the form, add a Button control in *exactly the same manner* as you added the label and text box. Set its Name property to **`btnOK`** and its Text property to **`&OK`**.



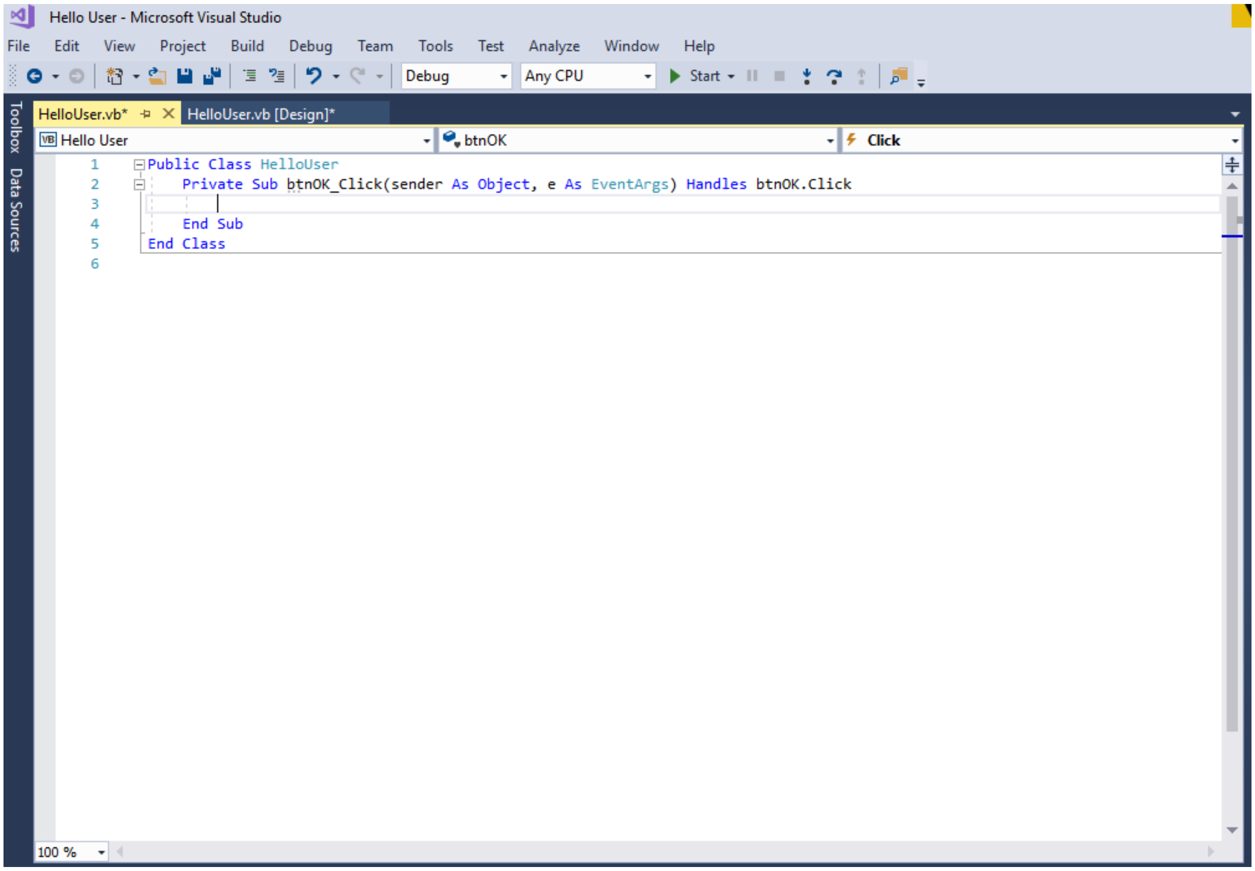
Button1 System.Windows.Forms.Button

Modifiers	**Friend**
⊞ Padding	0, 0, 0, 0
RightToLeft	No
⊞ Size	**75, 23**
TabIndex	**2**
TabStop	True
Tag	
Text	**&OK**
TextAlign	MiddleCenter

Text
The text associated with the control.

17. Now add a second Button control to the bottom right of the form by dragging the Button control from the Toolbox onto your form. Set the Name property to **`btnExit`** and the Text property to **`E&xit`**.

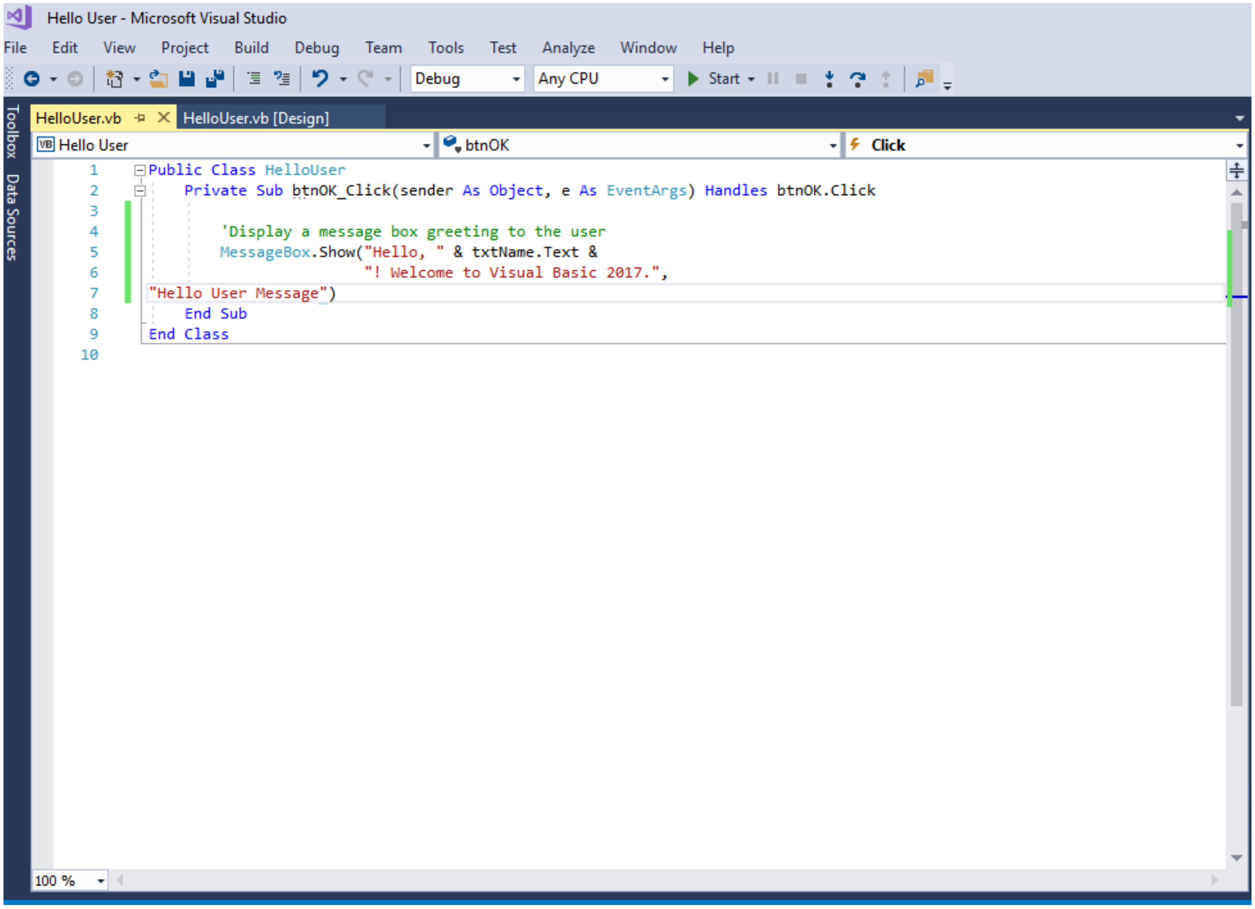
18. Now that you have the HelloUser form defined, you must add some code to make it actually do something interesting. To add the code for a control, you just double-click the control in question. Double click the OK button. This opens the Code Editor in the main window. Note that an additional tab had been created in the main window. Now you have the Design tab and the Code tab, each containing the name of the form you are working on.


Hello User - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Start
Toolbox
Data Sources
HelloUser.vb*
HelloUser.vb [Design]*
Hello User
btnOK
Click
100 %

```
Public Class HelloUser
    Private Sub btnOK_Click(sender As Object, e As EventArgs) Handles btnOK.Click

    End Sub
End Class
```

19. Type the following code EXACTLY as it appears below (note Visual Studio will automatically color code it for you). Once you have entered that code, click the Save All button.

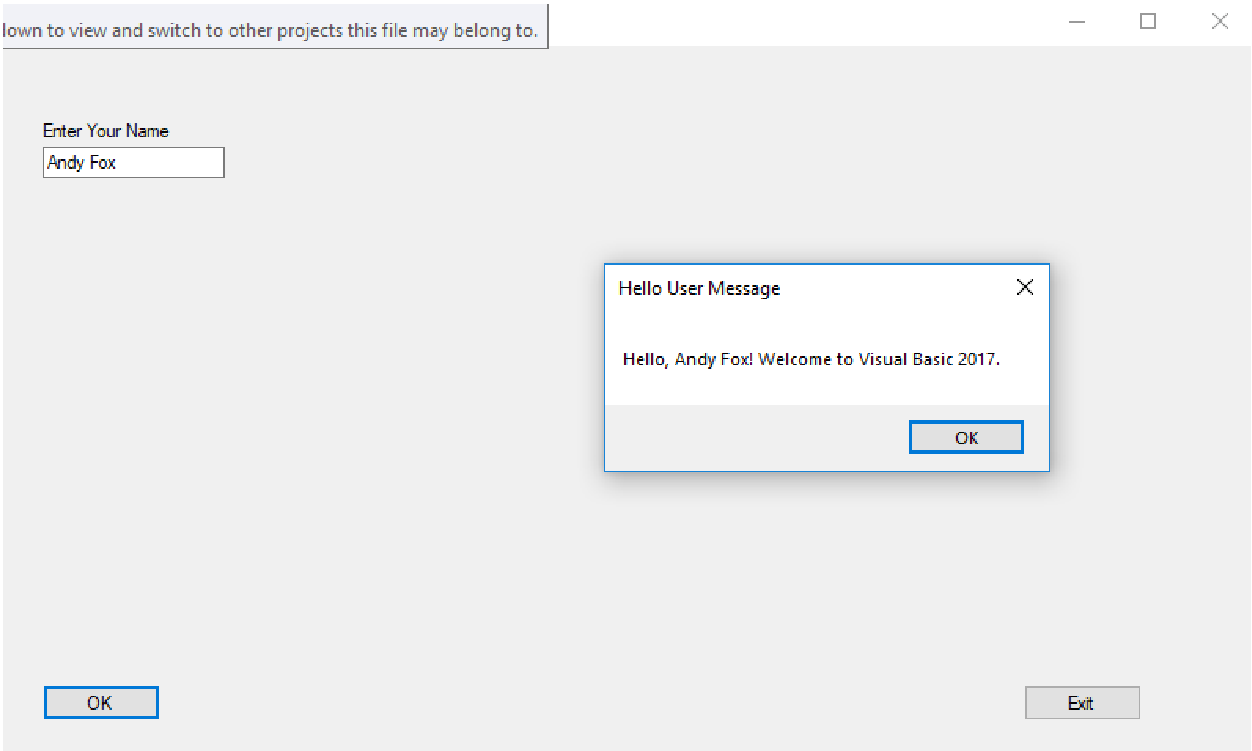

Hello User - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Start
HelloUser.vb
HelloUser.vb [Design]
Hello User
btnOK
Click
Toolbox
Data Sources
100 %

```
Public Class HelloUser
    Private Sub btnOK_Click(sender As Object, e As EventArgs) Handles btnOK.Click

        'Display a message box greeting to the user
        MessageBox.Show("Hello, " & txtName.Text &
                        "! Welcome to Visual Basic 2017.",
"Hello User Message")
    End Sub
End Class
```

20. Once you have saved, click the green arrow to test the code thus far. When the program comes up, enter your name and click OK. You should see the form below (but with your name instead of mine). Once Mr. Fox has checked it, you may close the form by clicking the close button in the upper right-hand corner (the Exit button doesn't work yet).

CHECK #2: CALL MR. FOX OVER AND SHOW HIM THAT YOU'VE REACHED THIS POINT (YOUR SCREEN SHOULD LOOK LIKE THE SCREEN ABOVE).



lown to view and switch to other projects this file may belong to.

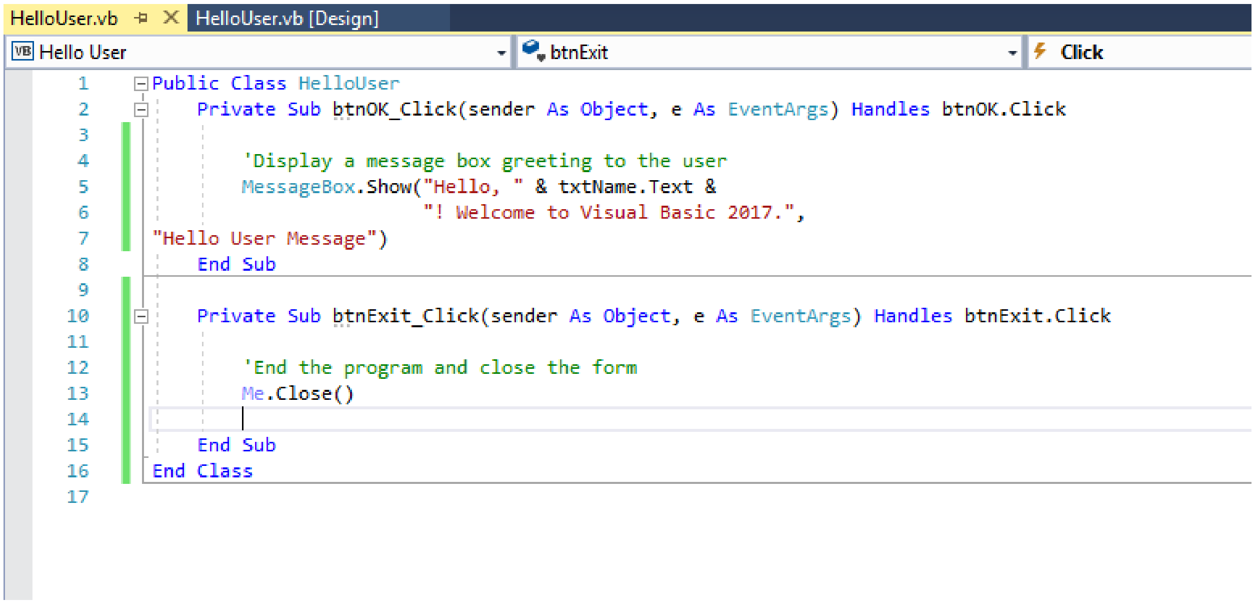
Enter Your Name: Andy Fox

Hello User Message

Hello, Andy Fox! Welcome to Visual Basic 2017.

Once Mr. Fox has checked your work, proceed to the next step.

21. Once you have closed it. Go back to the design tab (where you drew your form) and double click the Exit button. The add the following code and click Save All:



```
Public Class HelloUser
    Private Sub btnOK_Click(sender As Object, e As EventArgs) Handles btnOK.Click

        'Display a message box greeting to the user
        MessageBox.Show("Hello, " & txtName.Text &
                        "! Welcome to Visual Basic 2017.",
"Hello User Message")
    End Sub

    Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles btnExit.Click

        'End the program and close the form
        Me.Close()

    End Sub
End Class
```

22. Now click the Start Button again and let Mr. Fox know. He will come over and you can demonstrate the Exit button’s functionality.

CHECK #3: CALL MR. FOX OVER AND SHOW HIM THAT YOU’VE REACHED THIS POINT. THIS IS THE FINAL POINT. CONGRATULATIONS!