Computer Workshop - III

S. No	Experiments
1.	Installing of .NET
2.	Study **features** of VB.Net and **overview** of VB.Net.
3.	Programming exercise on using **various data types** of VB.Net.
4.	Programming exercise on **conversion of data types** in VB.Net.
5.	Write a program in VB.Net to **Add, Subtract** and **Multiply** two numbers.
6.	Write a program **to read names of three students and display it** on the screen.
7.	Write a Program using **PI as constant value** and **calculate area of circle**.
8.	Write a program to **display the first 10 natural numbers** and calculate their **Sum** and **Average** Value.
9.	Write a program using **enumerated data type** and assigning days of week from 1 to 7 and display their values.
10.	Write a program to find the **percentage** of students using obtained and total marks. Check whether a student is **Pass/Fail** using 40 as passing criteria.
11.	Write a program to input two strings and perform various string operations like **Concat, ToLower, ToUpper, Trim, Compare, Contains, Substrin**g etc.
12.	Write a program to read a single **dimensional array** of 20 numbers. Find & Display the **smallest and largest** of those numbers.
13.	Write a programs using conditional statements and loops: **Generate Fibonacci series**.
14.	Read **two matrices of 2 x 20**, **add** these matrices and display the resulting matrix.
15.	Write a program using a function to **reverse a number**.
16.	Write a sub procedure to **display the biggest of three numbers** passed as parameters.
17.	Write a program to declare a class of 'Box' having data members as height, length and breadth. Find and display the volume **of Box** using member functions of the class.
18.	Write a program to **create a form** by using tool box controls.
19.	Write a program for handling various mouse events such as **MouseHover, MouseDown** etc.
20.	Write a program for **File Handling** in VB.Net
21.	Write a program to design a **simple calculator**.

Note:

Experiment 1. Installing of **VB.NET**

To use Visual Studio 2015, you must have installed the following –

- Microsoft Visual Studio 2015 Update 3
- Microsoft .NET Core 1.0.1 - VS 2015 Tooling Preview 2

Microsoft provides a free version of visual studio which also contains the SQL Server and can be downloaded from

https://www.visualstudio.com/en-us/downloads/download-visual-studio-vs.aspx

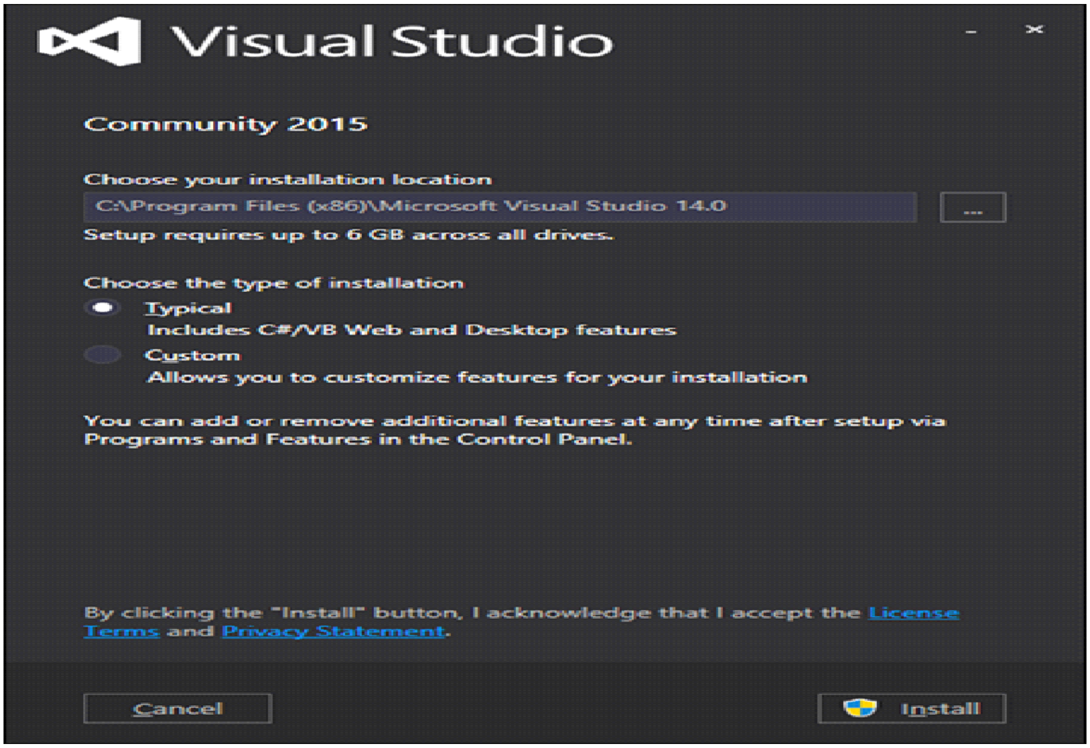
and Microsoft .NET Core 1.0.1 - VS 2015 Tooling Preview 2 can be downloaded from

https://www.visualstudio.com/downloads/

Installation of Visual Studio 2015

Follow these steps to install Visual Studio 2015 –

Step 1 – Once the downloading completes, then run the installer. The following dialog box will be displayed.



Visual Studio

Community 2015

Choose your installation location

C:\Program Files (x86)\Microsoft Visual Studio 14.0

Setup requires up to 6 GB across all drives.

Choose the type of installation

● Typical
Includes C#/VB Web and Desktop features

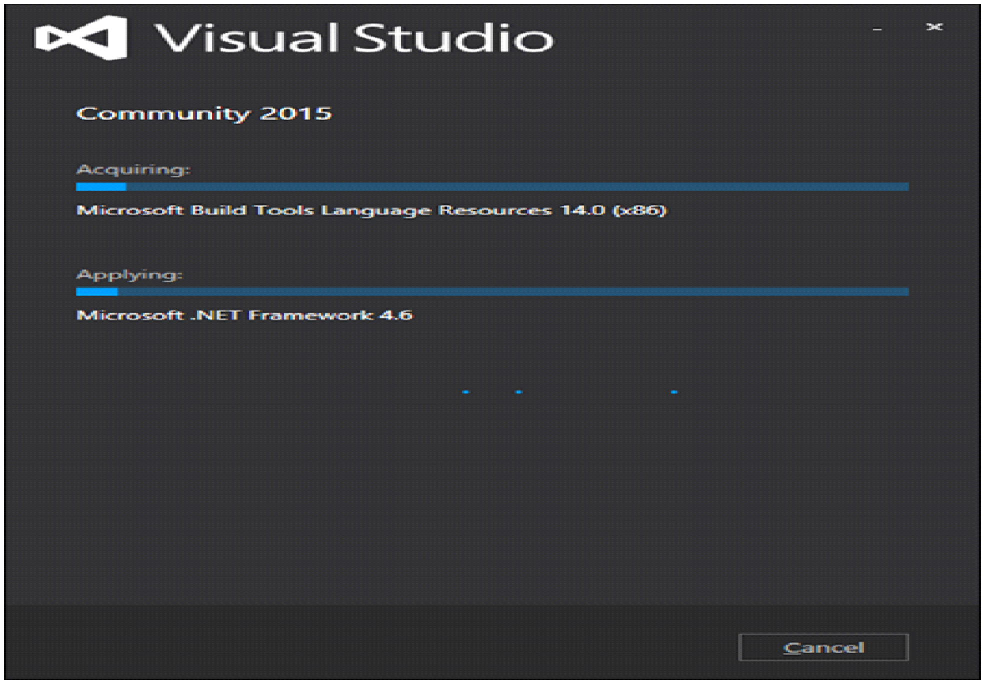
○ Custom
Allows you to customize features for your installation

You can add or remove additional features at any time after setup via Programs and Features in the Control Panel.

By clicking the "Install" button, I acknowledge that I accept the License Terms and Privacy Statement.

Note:

Step 2 – Click **Install** to start the installation process.



Visual Studio

Community 2015

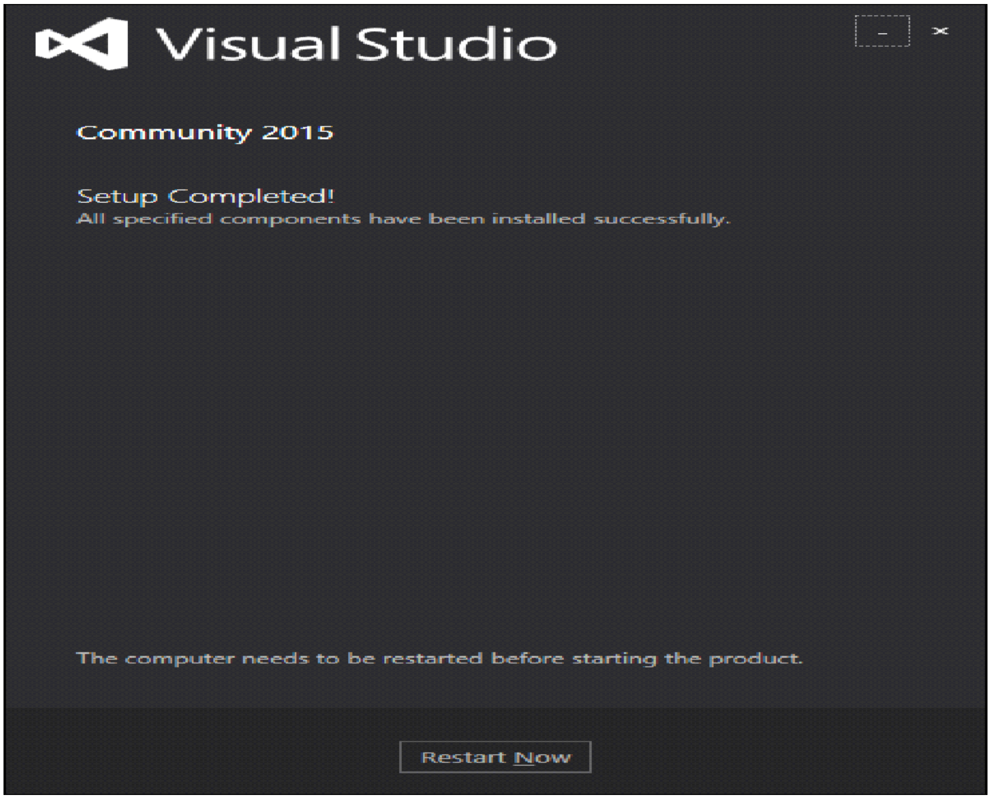
Acquiring:

Microsoft Build Tools Language Resources 14.0 (x86)

Applying:

Microsoft .NET Framework 4.6

Step 3 – Once the installation completes, you will see the following dialog box.



Visual Studio

Community 2015

Setup Completed!

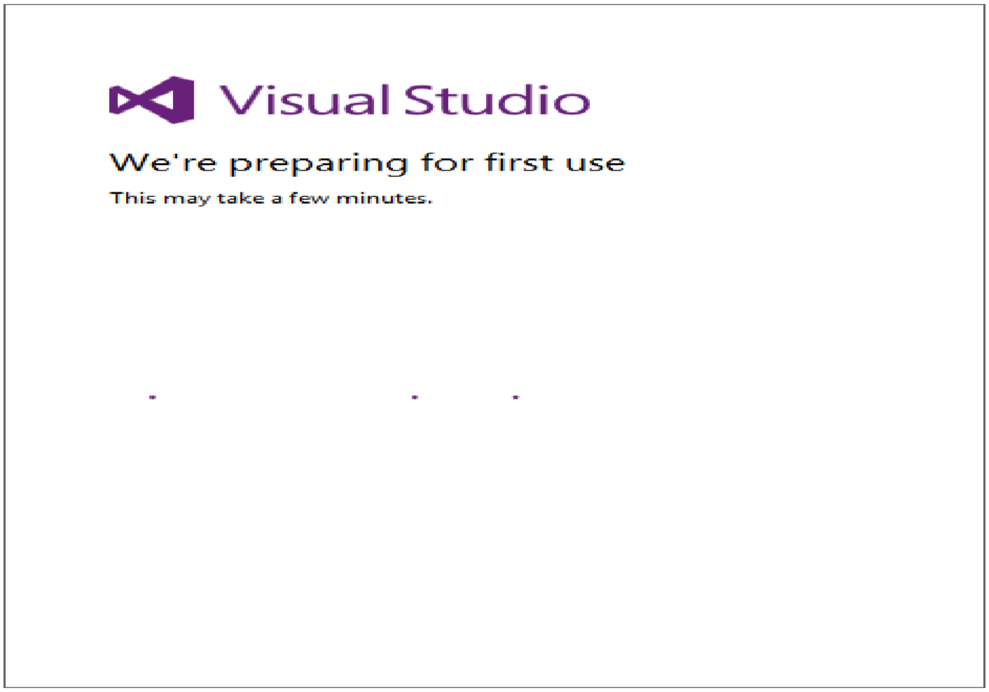
All specified components have been installed successfully.

The computer needs to be restarted before starting the product.

Note:

Step 4 – Close this dialog and restart your computer if required.

Step 5 – Open Visual Studio from the Start Menu; you will receive the following dialog box. It may take a few minutes to load and finally be used for the first time.

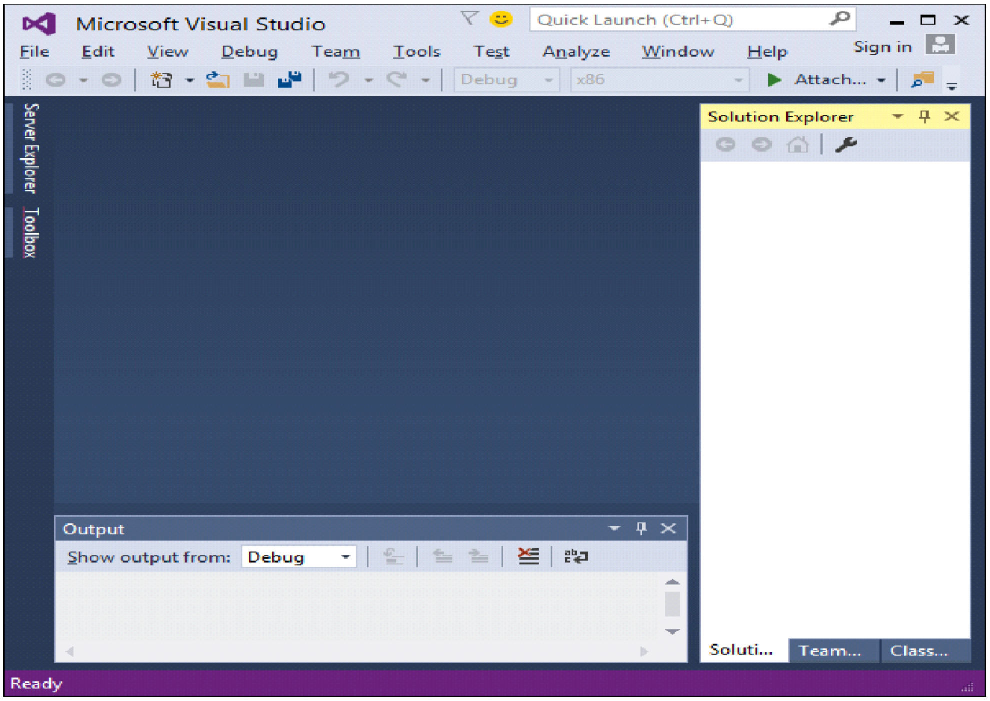


Visual Studio

We're preparing for first use

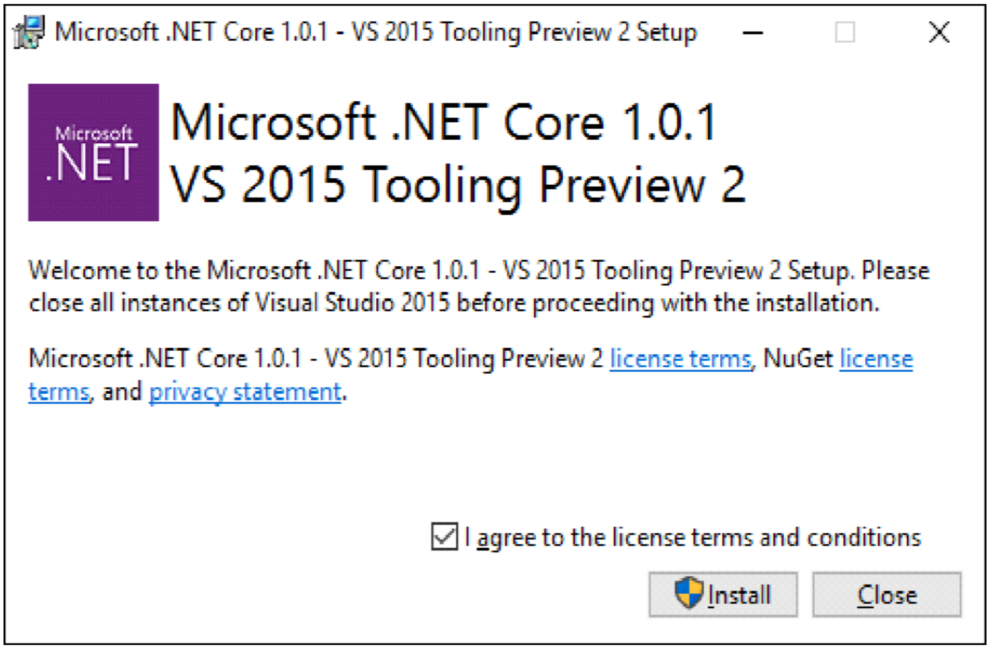
This may take a few minutes.

Step 6 – Once it is loaded, you will see the following screen.



Note:

Step 7 – Once Visual Studio installation is finished, then close Visual Studio and launch Microsoft .NET Core - VS 2015 Tooling Preview 2.



Microsoft .NET Core 1.0.1 - VS 2015 Tooling Preview 2 Setup

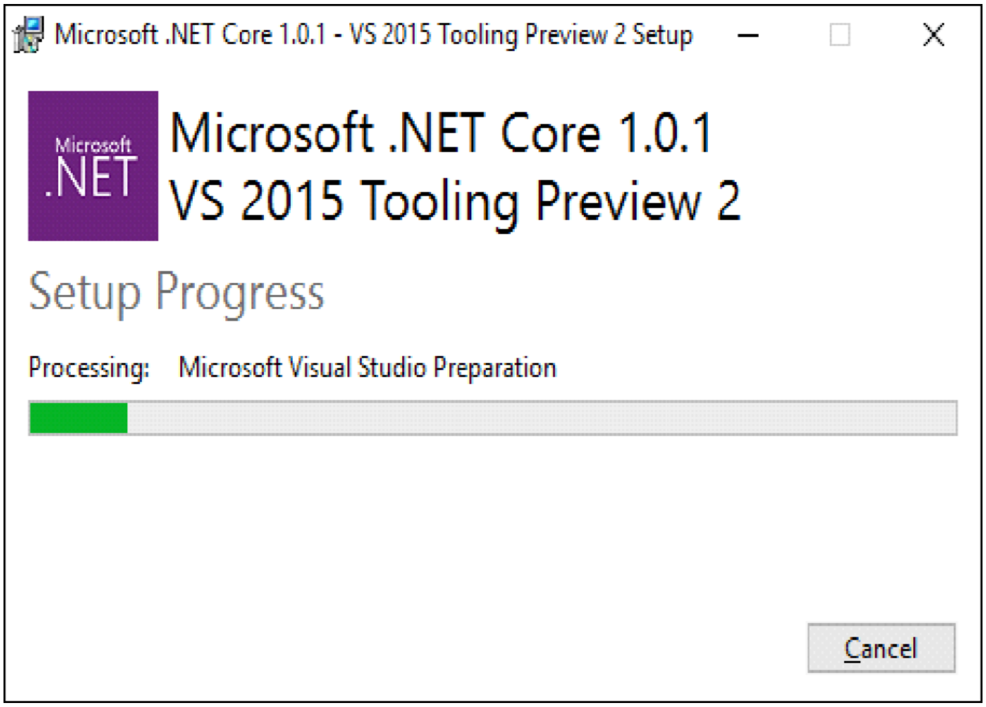
Microsoft .NET Core 1.0.1
VS 2015 Tooling Preview 2

Welcome to the Microsoft .NET Core 1.0.1 - VS 2015 Tooling Preview 2 Setup. Please close all instances of Visual Studio 2015 before proceeding with the installation.

Microsoft .NET Core 1.0.1 - VS 2015 Tooling Preview 2 license terms, NuGet license terms, and privacy statement.

☑ I agree to the license terms and conditions

Step 8 – Check the checkbox and click Install.



Microsoft .NET Core 1.0.1 - VS 2015 Tooling Preview 2 Setup

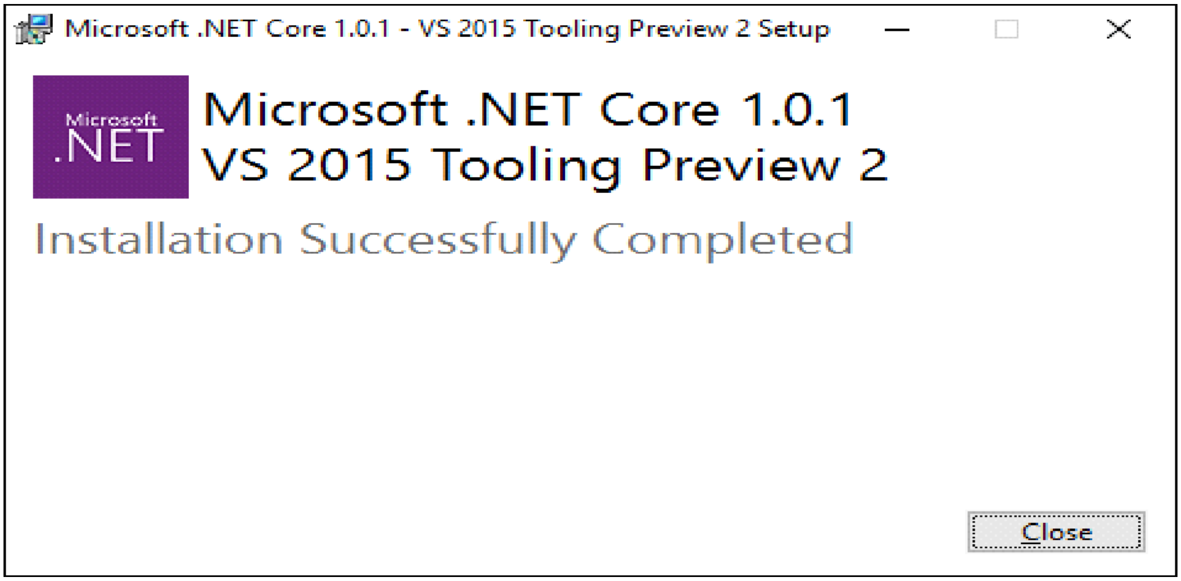
Microsoft .NET Core 1.0.1
VS 2015 Tooling Preview 2

Setup Progress

Processing: Microsoft Visual Studio Preparation

Note:

Step 9 — Once the installation completes, you will see the following dialog box.



Microsoft .NET Core 1.0.1 - VS 2015 Tooling Preview 2 Setup
Microsoft .NET
Microsoft .NET Core 1.0.1
VS 2015 Tooling Preview 2
Close

Installation Successfully Completed

You are now ready to start your application using .NET Core.

Experiment 2. Study **overview** of VB.Net and **features** of VB.Net.

Visual Basic .NET (VB.NET) is an object-oriented computer programming language implemented on the .NET Framework. Everything in VB.NET is an object, including all of the primitive types (Short, Integer, Long, String, Boolean, etc.) and user-defined types, events, and even assemblies. All objects inherit from the base class Object.

Features VB.Net

- Boolean Conditions
- Automatic Garbage Collection
- Standard Library
- Assembly Versioning
- Properties and Events
- Delegates and Events Management
- Easy-to-use Generics
- Indexers
- Conditional Compilation
- Simple Multithreading

Note:

Experiment 3. Programming exercise on using **various data types** of VB.Net.

Data Type	Storage Allocation	Value Range
Boolean	Depends on implementing platform	**True** or **False**
Byte	1 byte	0 through 255 (unsigned)
Char	2 bytes	0 through 65535 (unsigned)
Date	8 bytes	0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
Decimal	16 bytes	0 through +/- 79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) with no decimal point; 0 through +/- 7.9228162514264337593543950335 with 28 places to the right of the decimal
Double	Double	1.79769313486231570E+308 through -4.94065645841246544E-324, for negative values 4.94065645841246544E-324 through 1.79769313486231570E+308, for positive values
Integer	4 bytes	-2,147,483,648 through 2,147,483,647 (signed)
Long	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807(signed)
Object	4 bytes on 32-bit platform 8 bytes on 64-bit platform	Any type can be stored in a variable of type Object

Note:

SByte	1 byte	-128 through 127 (signed)
Short	2 bytes	-32,768 through 32,767 (signed)
Single	4 bytes	-3.4028235E+38 through -1.401298E-45 for negative values; 1.401298E-45 through 3.4028235E+38 for positive values
String	Depends on implementing platform	0 to approximately 2 billion Unicode characters
UInteger	4 bytes	0 through 4,294,967,295 (unsigned)
ULong	8 bytes	0 through 18,446,744,073,709,551,615 (unsigned)
User-Defined	Depends on implementing platform	Each member of the structure has a range determined by its data type and independent of the ranges of the other members
UShort	2 bytes	0 through 65,535 (unsigned)

```
Imports System

Module DataTypes
   Sub Main()
      Dim b As Byte
      Dim n As Integer
      Dim si As Single
      Dim d As Double
      Dim da As Date
      Dim c As Char
      Dim s As String
      Dim bl As Boolean
```

Note:

```
        b = 1
        n = 1234567
        si = 0.12345678901234566
        d = 0.12345678901234566
        da = Today
        c = "U"c
        s = "Me"
        Console.Write(c & " and," & s & vbCrLf)
            Console.WriteLine("declaring on the day of: {0}", da)
            Console.WriteLine("We will learn VB.Net seriously")
            Console.WriteLine("Lets see what happens to the floating point
variables:")
            Console.WriteLine("The Single: {0}, The Double: {1}", si, d)

        Console.ReadKey()
    End Sub

End Module
```

Experiment 4. Programming exercise on **conversion of data types** in VB.Net.

```
Imports System

Module DataTypes
    Sub Main()
        Dim n As Integer
        Dim da As Date
        Dim bl As Boolean = True
        n = 1234567
        da = Today

        Console.WriteLine(bl)
        Console.WriteLine(CSByte(bl))
        Console.WriteLine(CStr(bl))
        Console.WriteLine(CStr(da))
        Console.WriteLine(CChar(CChar(CStr(n))))
```

Note:

```
        Console.WriteLine(CChar(CStr(da)))
        Console.ReadKey()
    End Sub
End Module
```

Experiment 5. Write a program in VB.Net to **Add, Subtract** and **Multiply** two numbers.

```
Module Module1
    Sub Main()
        Dim a As Integer
        Dim b As Integer
        a = 10
        b = 5
        'Sum of a And be
        Console.WriteLine("Sum of two numbers: " & (a + b))
        'Submission of two number
        Console.WriteLine("Submission of two numbers: " & (a –
        b))
        'Multiplication of two number
        Console.WriteLine("Mul of two numbers: " & (a * b))
        Console.ReadLine()
    End Sub
End Module
```

Experiment 6. Write a program **to read names of three students and display it** on the screen.

```
Module Module1

    Sub Main()
        'Create a list of students.
        Dim students = GetStudents()
        'Display the names in the list.
        DisplayList(students)
                Console.ReadLine()
    End Sub
```

Note:

```
'Call DisplayList to see the names of the students in the list.
Sub DisplayList(ByVal studentCol As IEnumerable(Of Student))
    For Each st As Student In studentCol
        Console.WriteLine("First Name: " & st.First)
        Console.WriteLine(" Last Name: " & st.Last)
        Console.WriteLine()
    Next
End Sub

'Function GetStudents returns a list of Student objects.
Function GetStudents() As IEnumerable(Of Student)
    Return New List(Of Student) From
        {
         New Student("Ram", "Kumar", "Primary", 10),
         New Student("Amit", "Kumar", "HS", 2),
         New Student("Deepak", "Kumar", "UG", 7)
        }
End Function

'Each student has a first name, a last name, a class year, and
'a rank that indicates academic ranking in the student body.
Public Class Student
    Public Property First As String
    Public Property Last As String
    Public Property Year As String
    Public Property Rank As Integer

    Public Sub New()
    End Sub

    Public Sub New(ByVal firstName As String,
                   ByVal lastName As String,
                   ByVal studentYear As String,
                   ByVal studentRank As Integer)
        First = firstName
        Last = lastName
```

```
            Year = studentYear
            Rank = studentRank
        End Sub
    End Class
End Module
```

Experiment 7. Write a Program using **PI as constant value** and **calculate area of circle**.

```
Module Module1
    Public Sub Main(args() As String)
        Dim r As Integer = 22
        Dim a As Double
        Const PI = 3.14
        a = PI * r * r
        Console.WriteLine("Area of circle " & (a))
    End Sub
End Module
```

Experiment 8. Write a program to **display the first 10 natural numbers** and calculate their **Sum** and **Average** Value.

```
Imports System

Module Program
    Sub Main(args As String())
        Dim number As Integer = 11
        Dim sum As Integer = 0
        For i As Integer = 1 To number
            sum += i
        Next
        Dim Avg As Integer
        Avg = sum / number
        Console.WriteLine("Sum Of natural number" & Avg)
        Console.WriteLine("Sum Of natural number" & sum)
    End Sub
End Module
```

Experiment 9. Write a program using **enumerated data type** and assigning days of week from 1 to 7 and display their values. Days of week from 1 to 7 and display their values.

```
Imports System

Module Enum_Day
    Enum Weekday 'Enumeration name
        Monday = 1
        Tuesday = 2
        Wedneday = 3
        Thursday = 4
        Friday = 5
        Saturday = 6
        Sunday = 7
    End Enum
    Sub Main()
        Dim x As Integer = CInt(Weekday.Monday)
        Dim y As Integer = CInt(Weekday.Tuesday)
        Dim p As Integer = CInt(Weekday.Wedneday)
        'Console.WriteLine("Week Days name is {0}", Weekday.Monday)
        Console.WriteLine(" Value is  " & Weekday.Monday)
        Console.ReadKey()
 End Sub
End Module
```

Experiment 10. Write a program to find the **percentage** of students using obtained and total marks. Check whether a student is **Pass/Fail** using 40 as passing criteria.

```
Imports System

Module Module1
    Sub Main()
```

```
        Dim m1, m2, m3, m4, m5, total As Integer
        Dim name As String
        Dim avg As Double
        Console.WriteLine("Enter the name:")
        name = Console.ReadLine()
        Console.WriteLine("Enter the Marks:")
        m1 = Console.ReadLine()
        m2 = Console.ReadLine()
        m3 = Console.ReadLine()
        m4 = Console.ReadLine()
        m5 = Console.ReadLine()
        total = m1 + m2 + m3 + m4 + m5
        avg = total / 5
        Console.WriteLine("Total Marks=" & total)
        Console.WriteLine("Average=" & avg)

        'divide avg by 10 to make the calculation easier
        avg = avg / 10

        If (avg >= 4) Then
            Console.WriteLine("Pass")
        Else
            Console.WriteLine("Fail")
        End If

        Console.ReadKey()
    End Sub
End Module
```

Experiment 11. Write a program to input two strings and perform various string operations like **Concat, ToLower, ToUpper, Trim, Compare, Contains, Substrin**g etc.

```
Imports System
Module strings
    Sub Main()
        Dim fname, lname, fullname, greetings As String
        fname = "Saurav Kumar"
```

Note:

```
        lname = "Pandey"
        fullname = fname + " " + lname
        Console.WriteLine("Full Name: {0}", fullname)

        'By using string constructor
        Dim letters As Char() = {"H", "e", "l", "l", "o"}
        greetings = New String(letters)
        Console.WriteLine("Greetings: {0}", greetings)

        'Methods returning String
        Dim array() As String = {"Hello", "From", "Ambedkar", "DSEU",
    "Shakarpur", "Campus-I"}
        Dim message As String = String.Join(" ", sarray)
        Console.WriteLine("Message: {0}", message)

        'Method Compare String
        Dim str1, str2 As String
        str1 = "This is text"
        str2 = "This is text"

        If (String.Compare(str1, str2) = 0) Then
            Console.WriteLine(str1 + " and " + str2 + " are equal.")
        Else
            Console.WriteLine(str1 + " and " + str2 + " are not equal.")
        End If

        'Method to Find a given word in String
        Dim str3 As String
        str3 = "This is test"

        If (str3.Contains("test")) Then
            Console.WriteLine("The sequence 'test' was found.")
        End If

        'Method to Concat String
        Dim str As String
        str = "    Last night I was in theatre     "
```

Note:

```
        Console.WriteLine(str)

        Dim substr As String = str.Substring(23)
        Console.WriteLine(substr)

        'Method toLower
        Dim lower As String = str.ToLower()
        Console.WriteLine(lower)

        'Method ToUpper
        Dim upper As String = str.ToUpper()
        Console.WriteLine(upper)

        'Method Concat
        Dim result As String = String.Concat(str1, str2)
        Console.WriteLine(result)

        'Method Trim
        Dim result1 As String = str.Trim()
        Console.WriteLine(result1)

    End Sub
End Module
```

Experiment 12. Write a program to read a single **dimensional array** of 20 numbers. Find & Display the **smallest and largest** of those numbers.

```
Imports System
Module IterateArray
    Public Sub Main()
        Dim numbers = {10, 20, 30, 40, 4, 5, 6, 7, 8, 9, 2, 43, 5, 6, 75, 5, 65, 45, 66,
5}
        Dim max = numbers(0)
        Dim min = numbers(0)
        Console.WriteLine(max)
        For index = 0 To numbers.GetUpperBound(0)
            If numbers(index) > max Then
        max = numbers(index)
      End If
      If numbers(index) < min Then
        min = numbers(index)
      End If
      Console.Write(" " & numbers(index))
    Next
    Console.WriteLine(" " & max)
    Console.WriteLine(" " & min)
  End Sub
End Module
```

Experiment 13. Write a program using conditional statements and loops:

A.) Generate Fibonacci series.

```
Imports System
Module Module1
 Sub Main ()
    Dim a, b, c, n, i As Integer
    Console.Write ("Enter how many elements :-")
    n = Val (Console.ReadLine())
    a = 0
    b = 1
    Console.Write(" " & a)
    Console.Write(" " & b)
    i = 1
    While (i < n - 1)
      c = a + b
      Console.Write(" " & c)
      a = b
      b = c
      i = i + 1
    End While
    Console.ReadLine()
 End Sub

End Module
```

Note:

Experiment 14. Read **two matrices of 2 x 2**, **add** these matrices and display the resulting matrix.

```
Imports System

Module Module1

Sub Main ()
    Dim matrix1(,) As Integer = New Integer(2, 2) {}
    Dim matrix2(,) As Integer = New Integer(2, 2) {}
    Dim matrix3(,) As Integer = New Integer(2, 2) {}

    Console.WriteLine("Enter Matrix1: ")
    For i = 0 To 1 Step 1
      For j = 0 To 1 Step 1
        Console.Write("Enter element[{0}][{1}]: ", i, j)
        matrix1(i, j) = Integer.Parse(Console.ReadLine())
      Next
    Next
    Console.WriteLine("Enter Matrix2: ")
    For i = 0 To 1 Step 1
      For j = 0 To 1 Step 1
        Console.Write("Enter element[{0}][{1}]: ", i, j)
        matrix2(i, j) = Integer.Parse(Console.ReadLine())
      Next
    Next
      'Add Matrix1 and Matrix2
    For i = 0 To 1 Step 1
      For j = 0 To 1 Step 1
        matrix3(i, j) = matrix1(i, j) + matrix2(i, j)
      Next
    Next
    Console.WriteLine("Matrix1: ")
    For i = 0 To 1 Step 1
      For j = 0 To 1 Step 1
        Console.Write("{0} ", matrix1(i, j))
      Next
```

Note:

```
            Console.WriteLine()
        Next
        Console.WriteLine("Matrix2: ")
        For i = 0 To 1 Step 1
            For j = 0 To 1 Step 1
                Console.Write("{0} ", matrix2(i, j))
            Next
            Console.WriteLine()
        Next
        Console.WriteLine("Addition of Matrix1 and Matrix2: ")
        For i = 0 To 1 Step 1
            For j = 0 To 1 Step 1
                Console.Write("{0} ", matrix3(i, j))
            Next
            Console.WriteLine()
        Next
    End Sub

End Module
```

Experiment 15. Write a program using a function to **reverse a number**.

```
Module Module1

    Sub Main()
        Dim number As Integer = 0
        Dim remainder As Integer = 0
        Dim reverse As Integer = 0

        Console.Write("Enter the number: ")
        number = Integer.Parse(Console.ReadLine())

        While (number > 0)
            ‘Console.WriteLine (" number before mod" & number)
            remainder = number Mod 10
            Console.WriteLine(" number after mod" & number)
            Console.WriteLine(" remainder" & remainder)
            reverse = reverse * 10 + remainder
            Console.WriteLine(" reverse" & reverse)
            number = number / 10
        End While

        Console.WriteLine("Reverse: {0}", reverse)
    End Sub
End Module
```

Experiment 16. Write a sub procedure to **display the biggest of three numbers** passed as parameters.

```
Imports System

Module paramByval
    Dim large As Integer
    Sub largest(ByVal x As Integer, ByVal y As Integer, ByVal z As Integer)

        If x > y Then

        ElseIf x > z Then
            large = x
        ElseIf z > y Then
            large = z
        Else
            large = y
        End If

    End Sub
    Sub Main ()
        ‘Local variable definition
        Dim a As Integer = 100
        Dim b As Integer = 200
        Dim c As Integer = 300
        Console.WriteLine("Before operation, value of a : {0}", a)
        Console.WriteLine("Before operation, value of b : {0}", b)
        Console.WriteLine("Before operation, value of a : {0}", c)
```

Note:

```
        'Calling a function to find the large number'
        largest (a, b, c)
        Console.WriteLine("After operation, value of a : {0}", a)
        Console.WriteLine("After operation, value of b : {0}", b)
        Console.WriteLine("After operation, value of b : {0}", c)
        Console.WriteLine(large)
        Console.ReadLine()
    End Sub
End Module
```

Experiment 17. Write a program to declare a class of ‘Box’ having data members as height, length and breadth. Find and display the volume **of Box** using member functions of the class.

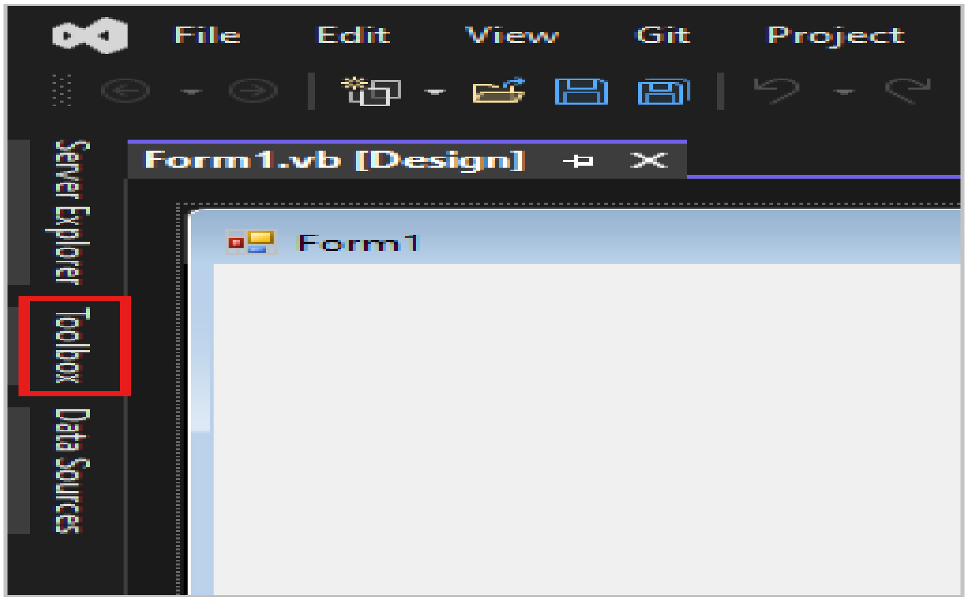
```
Imports System
Module mybox
    Class Box
        Public length As Double   ' Length of a box
        Public breadth As Double   ' Breadth of a box
        Public height As Double   ' Height of a box
        Public Sub setLength(ByVal len As Double)
            length = len
        End Sub
        Public Sub setBreadth(ByVal bre As Double)
            breadth = bre
        End Sub
        Public Sub setHeight(ByVal hei As Double)
            height = hei
        End Sub
        Public Function getVolume() As Double
            Return length * breadth * height
        End Function
    End Class
    Sub Main()
        Dim Box1 As Box = New Box()      ' Declare Box1 of type Box

        Dim volume As Double = 0.0    ' Store the volume of a box here
        ' box 1 specification
        Box1.setLength(12.0)
        Box1.setBreadth(7)
        Box1.setHeight(9.0)

        ' volume of box 1
        volume = Box1.getVolume()
        Console.WriteLine("Volume of Box1 : {0}", volume)


        Console.ReadKey()
    End Sub
End Module
```

Experiment 18. Write a program to **create a form** by using tool box controls.

Add a button to the form

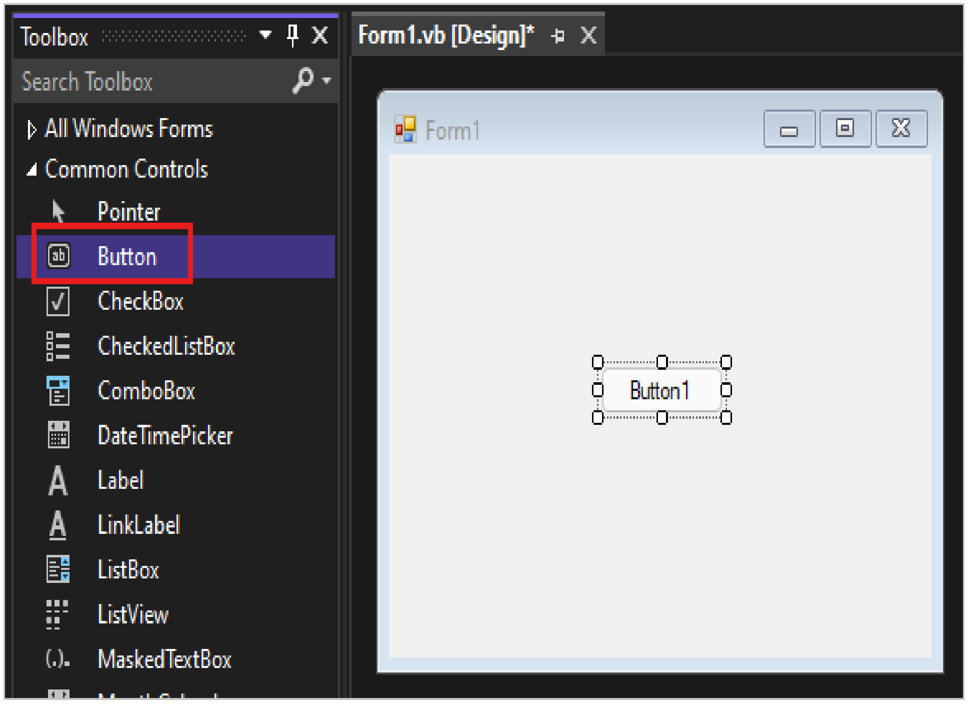
After you select your Visual Basic project template and name your file, Visual Studio opens a form for you. A form is a Windows user interface. You'll create a "Hello World" application by adding controls to the form.

1. On the left side of the Visual Studio IDE, select the Toolbox tab. If you don't see it, select View > Toolbox from the menu bar or Ctrl+Alt+X.



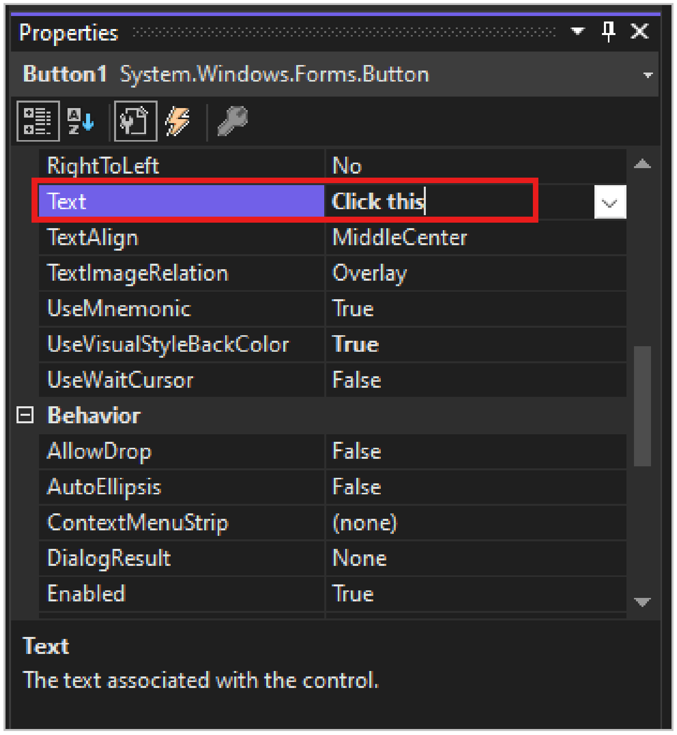
If you want, select the Pin icon to dock the Toolbox window.

2. Select the Button control and then drag it onto the form.



Note:

3. In the Appearance section of the Properties window, for Text, type *Click this*, and then press Enter.



Properties

Button1 System.Windows.Forms.Button

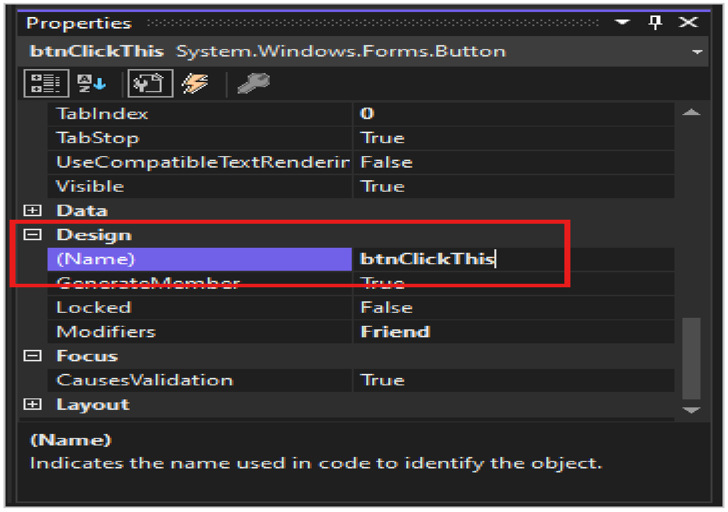
RightToLeft	No
Text	Click this
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseMnemonic	True
UseVisualStyleBackColor	**True**
UseWaitCursor	False
⊟ **Behavior**	
AllowDrop	False
AutoEllipsis	False
ContextMenuStrip	(none)
DialogResult	None
Enabled	True

Text

The text associated with the control.

If you don't see the Properties window, you can open it from the menu bar. Select View > Properties Window or press F4.

4. In the Design section of the Properties window, change the name from Button1 to *btnClickThis*, and then press Enter.



Properties

btnClickThis System.Windows.Forms.Button

TabIndex	**0**
TabStop	True
UseCompatibleTextRenderir	False
Visible	True
⊞ **Data**	
⊟ **Design**	
(Name)	btnClickThis
GenerateMember	True
Locked	False
Modifiers	**Friend**
⊟ **Focus**	
CausesValidation	True
⊞ **Layout**	

(Name)

Indicates the name used in code to identify the object.

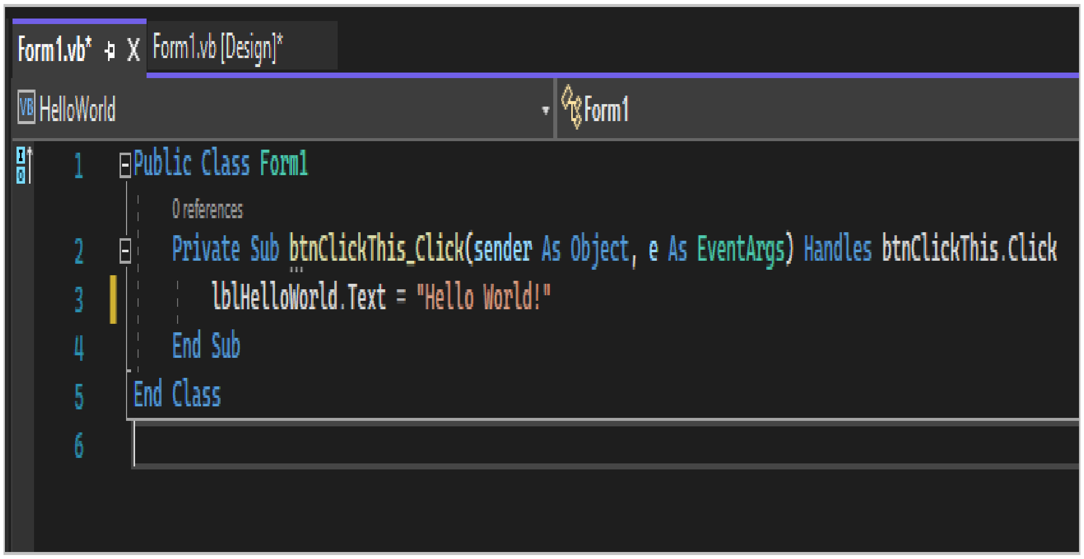
Note:

Note
if you've alphabetized the list in the Properties window, Button1 appears in the (Data Bindings) section, instead.

Add a label and code

Now that you've added a button control to create an action, add a label control to send text to.

1. Select the Label control in the Toolbox window, and then drag it onto the form. Place it beneath the Click this button.
2. In either the Design section or the (DataBindings) section of the Properties window, change the name Label1 to *lblHelloWorld*, and then press Enter.
3. In the Form1.vb [Design] window, double-click the Click this button to open the Form1.vb window.
   Another option is to expand Form1.vb in Solution Explorer, and then select Form1.
4. In the Form1.vb window, between the Private Sub and End Sub lines, enter *lblHelloWorld.Text = "Hello World!"* as shown in the following screenshot:



```
Public Class Form1
    Private Sub btnClickThis_Click(sender As Object, e As EventArgs) Handles btnClickThis.Click
        lblHelloWorld.Text = "Hello World!"
    End Sub
End Class
```

Experiment 19. Write a program for handling various mouse events such as **MouseHover, MouseDown** etc.

a. Add three labels, three text boxes and a button control in the form.
b. Change the text properties of the labels to - Customer ID, Name and Address, respectively.
c. Change the name properties of the text boxes to txtID, txtName and txtAddress, respectively.
d. Change the text property of the button to 'Submit'.
e. Add the following code in the code editor window –

```
Public Class Form1
  Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    ‘Set the caption bar text of the form.
    Me.Text = "Hello"
  End Sub

  Private Sub txtID_MouseEnter(sender As Object, e As EventArgs)_
    Handles txtID.MouseEnter
    'Code for handling mouse enter on ID textbox
    txtID.BackColor = Color.CornflowerBlue
    txtID.ForeColor = Color.White
  End Sub

  Private Sub txtID_MouseLeave(sender As Object, e As EventArgs) _
    Handles txtID.MouseLeave
    'Code for handling mouse leave on ID textbox

    txtID.BackColor = Color.White
    txtID.ForeColor = Color.Blue
  End Sub
```

Note:

```
    Private Sub txtName_MouseEnter(sender As Object, e As EventArgs) _
        Handles txtName.MouseEnter
        'Code for handling mouse enter on Name textbox
        txtName.BackColor = Color.CornflowerBlue
        txtName.ForeColor = Color.White
    End Sub

    Private Sub txtName_MouseLeave(sender As Object, e As EventArgs) _
        Handles txtName.MouseLeave
        'Code for handling mouse leave on Name textbox
        txtName.BackColor = Color.White
        txtName.ForeColor = Color.Blue
    End Sub

    Private Sub txtAddress_MouseEnter(sender As Object, e As EventArgs) _
        Handles txtAddress.MouseEnter
        'Code for handling mouse enter on Address textbox
        txtAddress.BackColor = Color.CornflowerBlue
        txtAddress.ForeColor = Color.White
    End Sub

    Private Sub txtAddress_MouseLeave(sender As Object, e As EventArgs) _
        Handles txtAddress.MouseLeave
        'Code for handling mouse leave on Address textbox
        txtAddress.BackColor = Color.White
        txtAddress.ForeColor = Color.Blue
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) _
        Handles Button1.Click
        MsgBox("Thank you " & txtName.Text & ", for your kind cooperation")
    End Sub
End Class
```

Note:

Experiment 20. Write a program for **File Handling** in VB.Net

First make a text file with name Myfile

```
Imports System.IO
Module StReader
    Sub Main()

        Dim St As StreamReader = New StreamReader("C:\Users\Graphics Lab
1\Desktop\Myfile.txt")
        Dim ln As String

        ln = St.ReadLine()
        While (ln <> Nothing)
            Console.WriteLine(ln)
            ln = St.ReadLine()
        End While
        St.Close()
        Console.ReadKey()
    End Sub
End Module
```

Note: