

Supporting Australian Schools

Designed for Australian
Secondary Schools
Years 9 - 12

Visual Basic

An Introduction to programming

The image displays three screenshots of the Visual Basic IDE, illustrating the development of two different applications.

Top Screenshot: ShoppingList
The IDE shows a project named "ShoppingList" with a single file "Form1.vb". The code defines a public class "Form1" with a private sub "BtnAdd_Click" that handles the "btnAdd.Click" event. The code prompts the user for the number of items, then iterates through that number, prompting for item names and quantities. It calculates the total cost and displays it.

```
1 Public Class Form1
2     Private Sub BtnAdd_Click(sender As Object, e As EventArgs) Handles btnAdd.Click
3         Dim intSize As Integer = InputBox("How many items do you need?")
4         Dim Groceries(intSize - 1, 2) As String
5         Dim dblTotal As Double
6         For i = 0 To intSize - 1
7
8             Groceries(i, 0) = InputBox("Enter item: " & i + 1)
9             Groceries(i, 1) = InputBox("Enter Quantity: ")
10            Groceries(i, 2) = InputBox("Enter Price: ")
11            lbxShopping.Items.Add(Groceries(i, 0) & " : " & (Groceries(i, 1)) & " items : $" & (Groceries(i, 2)))
12            dblTotal = dblTotal + (Groceries(i, 1) * Groceries(i, 2))
13            lbxShopping.Items.Add("Total Cost: " & dblTotal)
14        Next
15    End Sub
16 End Class
17
```

Middle Screenshot: Records Example
The IDE shows a project named "Records Example" with a single file "Form1.vb". The code defines a private sub "BtnAddRecord_Click" that handles the "btnAddRecord.Click" event. It loads an XML document "PhoneBook.xml", iterates through the elements, and prompts the user for contact details. It then creates a new XML element "Phone_Contact" with the provided details and adds it to the root of the document.

```
52
53 Private Sub BtnAddRecord_Click(sender As Object, e As EventArgs) Handles btnAddRecord.Click
54     Dim xelement As XElement = XElement.Load("PhoneBook.xml")
55     Dim Phone_Book As IEnumerable(Of XElement) = xelement.Elements()
56     Dim Document As XDocument
57     Document = XDocument.Load("PhoneBook.xml")
58
59     Dim strName As String = txtName.Text
60     Dim strAddress As String = txtAddress.Text
61     Dim strPhone As String = txtPhone.Text
62     Dim strEmail As String = txtEmail.Text
63
64     Dim root = New XElement("Phone_Contact")
65     Dim NameElement = New XElement("Name", strName)
66     Dim AddressElement = New XElement("Address", strAddress)
67     Dim PhoneElement = New XElement("Phone", strPhone)
68     Dim EmailElement = New XElement("Email", strEmail)
69
70     root.Add(NameElement, AddressElement, PhoneElement, EmailElement)

```

Bottom Screenshot: Records Example
This screenshot shows the same "Records Example" project, but with the "Properties" window open at the bottom right, indicating the user is configuring the application's settings.

Table of Contents

Chapter 1 Setting Up Visual Studio Community	Page 3
Chapter 2 My First Program & GUI design	Page 6
Chapter 3 The Age & Time Calculators	Page 10
Chapter 4 Variables, Data Types and Control Structures	Page 13
Chapter 5 BMI Calculator	Page 16
Chapter 6 Validation & Nested IF Statements	Page 17
Chapter 7 GUI Objects & Pizza Ordering App	Page 19
PROJECT 1 COFFEE ORDERING APP	Page 24
Chapter 8 Iteration	Page 27
Chapter 9 The Guessing Game & Trace Tables	Page 29
Chapter 10 Functions	Page 32
PROJECT 2 TAX CALCULATOR APP	Page 33
Chapter 11 Data Structures	Page 34
Chapter 12 Arrays & Data Storage in Text Files	Page 38
PROJECT 3 TENNIS COMPETITION CALCULATOR	Page 43
Chapter 13 Records & Data Storage in XML	Page 45
Chapter 14 Login System using XML	Page 48
PROJECT 4 ATM CALCULATOR	Page 51
Chapter 15 Extra Tips & Tricks	Page 53
Chapter 16 Project Ideas	Page 56

This symbol indicates a working demonstration VB program is included on your USB or available on your www.TeachIT.com.au account.



ACKNOWLEDGEMENTS

Microsoft Visual Studio Community 2019 screen shots used with permission from Microsoft 2021

This publication is designed to meet the teaching and learning needs of teachers and students of Digital Technologies Australian Curriculum, Assessment and Reporting Authority 2015.

© Vic Farrell Publishing 2022

ISBN: 978-0-6486708-8-9

For extra teaching resources: teachit.net.au & vicfarrell.com.au

Chapter 1

Setting Up Visual Studio Community

Microsoft Visual Studio Community can be found on the Microsoft website:
<https://visualstudio.microsoft.com/vs/community/>

This book uses Visual Studio Community 2019, but the concepts and syntax used will still be applicable in later versions.

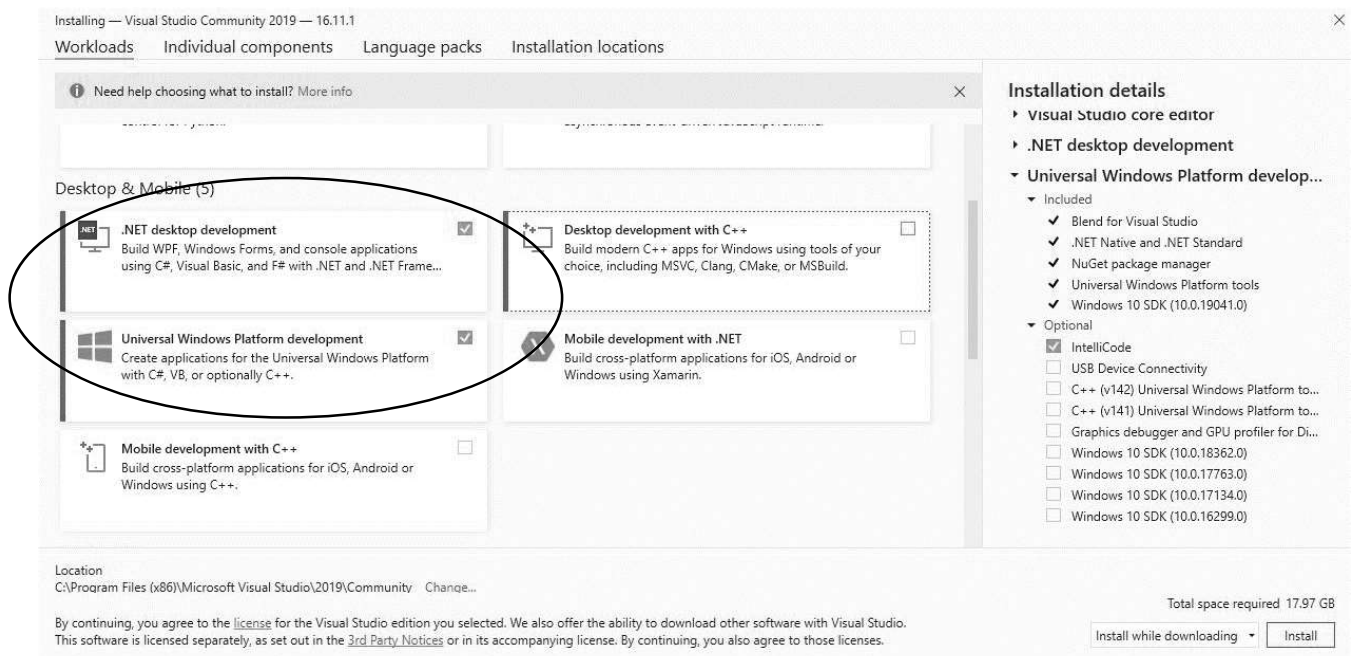
Visual Studio Community

A fully-featured, extensible, free IDE for creating modern applications for Android, iOS, Windows, as well as web applications and cloud services.

[Download Visual Studio](#) ↓

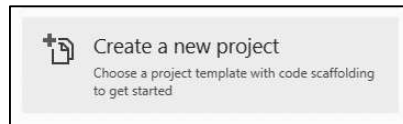
Click “Download Visual Studio ↓” and it will download an executable file. You will need a computer that is running Windows because it uses all the interface objects of the Windows library. If you have an Apple device, you can create a partition and run Windows.

Once you have downloaded the exe file, it will ask you for your Administration password to your computer. Then you need to click “Continue”. Once some of the software is downloaded a new window will open up. It gives you an option to choose a framework. It is advisable not to click too many of these options as it will take up a large amount of your hard drive space.

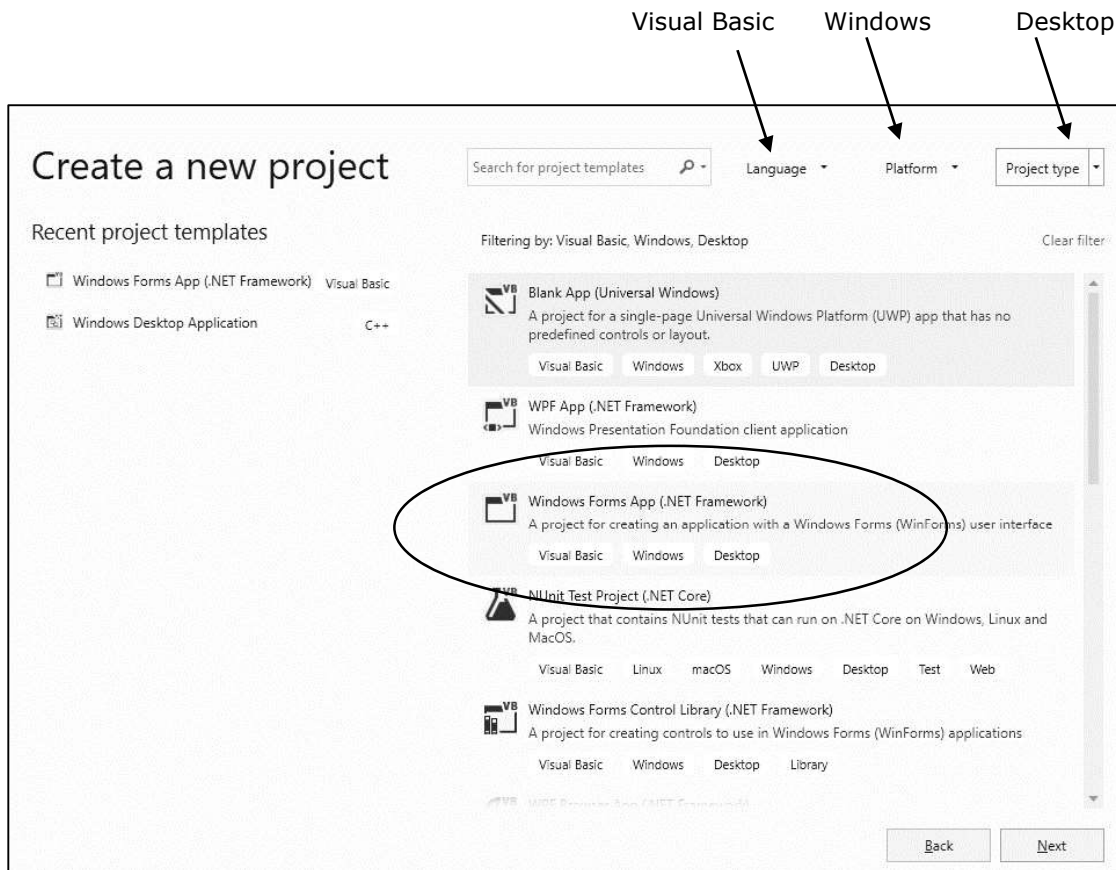


You can choose .NET desktop Development or Universal Windows Platform Development. Either of these frameworks will work for the activities in this book.

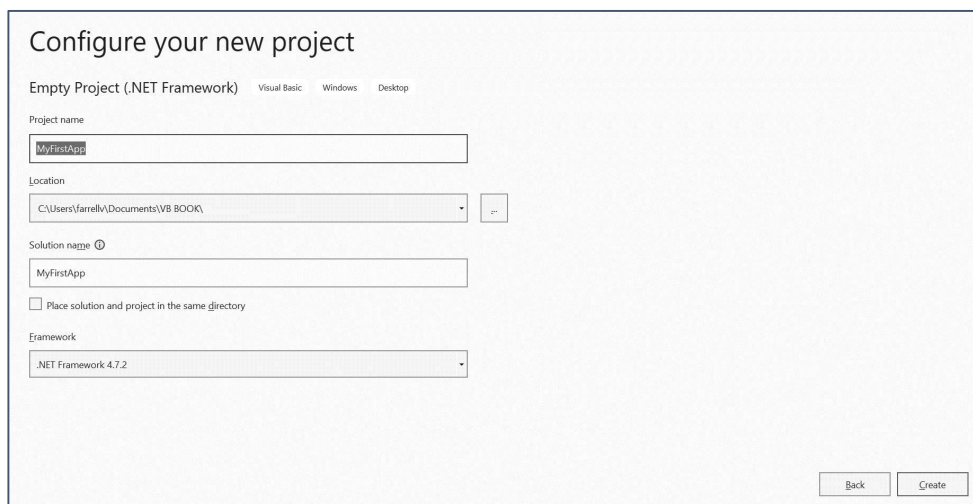
Once it is installed you can open Visual Studio and click on "Create a new project".



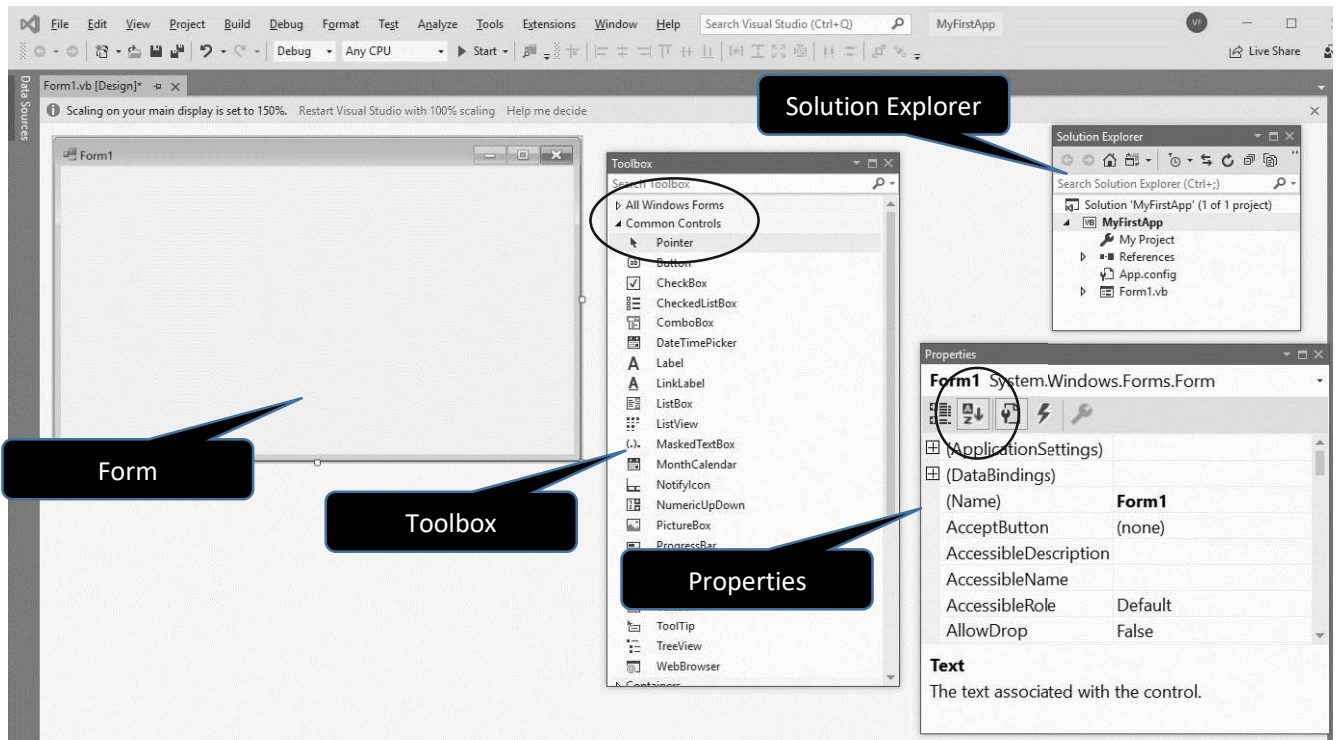
Now the next steps are VERY IMPORTANT!! You need to select Windows Forms App (.NET Framework). You can see them highlighted below. You can also search using the options at the top.



Now this is the step where everyone comes unstuck, so PAY ATTENTION! You need to NAME your application and identify the location where it will be stored. You will NOT be able to rename it or re-locate it easily. So pay attention! Let's call our first application "MyFirstApp" and put it in a folder called VB BOOK.



OK! We have Visual Studio Community Open now and you need to set it up like this...



If you can't find the Toolbox, the Property box or the Solution Explorer you will find them under the View menu. You will need each of these windows to set up your application.

Form - is where we are going to create our User Interface.

Toolbox - contains all the objects we need to put on our Form.

Properties - allows us to set properties for each object.

Solution Explorer - when you have many forms, you will use this window to locate them.

To make things easier for you to find everything you need make sure you only have "Common Controls" open in your Toolbox window and the a-z button selected in your Properties window. (See both circled above.)

Chapter 2

My First Program & GUI Design

Theory!

Let's talk about how programming works. There are three things every computer program does:

1. Reads in data,
2. Processes data,
3. Displays Data

If we were going to create a basic calculator that adds two number together it would first read in the two values, then add them together, then display the answer.

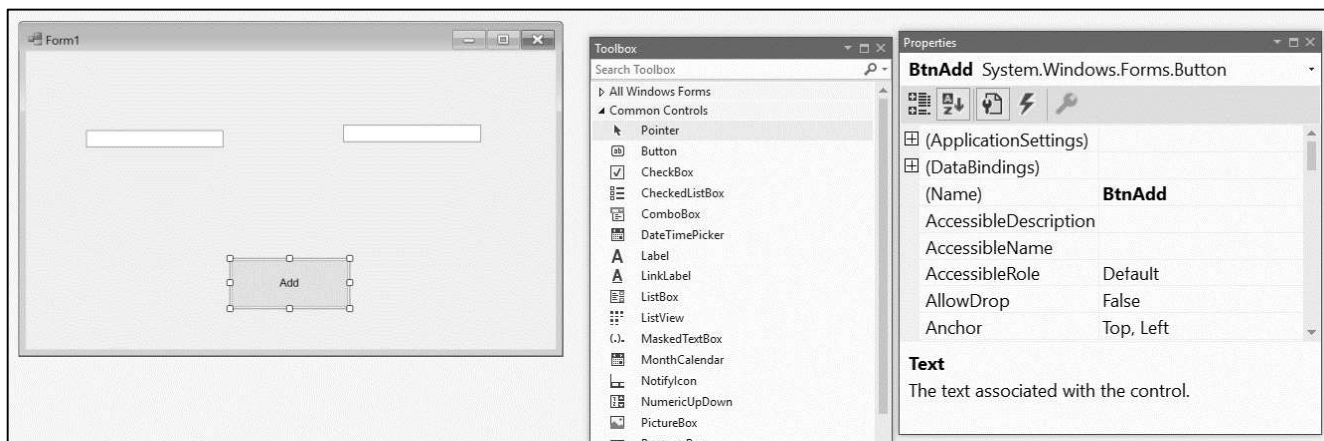
In VB we need to create our interface first, then we can code the instructions.

How to build an interface:

1. Go to the Toolbox and select "Textbox"
2. Click and drag on the Form and the textbox will appear on the Form.
3. Go to the Properties Window and find "(Name)". Let's name this textbox 'txtNumber1'
4. Create a second textbox and name it 'txtNumber2' (Don't use spaces when naming!)
5. Go to the Toolbox and select "Button" and name it 'btnAdd'
6. With the Button selected in the form, scroll in the Properties window to find "Text" and write 'Add'.

You will end up with an interface like the one below. All the items from the Toolbox are called OBJECTS! Objects form the interface and manage the data into and out of the program.

Don't forget: If you are finding it hard to navigate the Toolbox and Properties window click on the circled settings. Common Controls in the Toolbox are all you need and the A-Z button puts the (Name) property at the top and everything else in alphabetical order



You have completed your building your interface now let's code a solution. To access the window where we can start writing our code, double click the object that will execute the code. (*That's technical talk for double click the button you created on your form.*)

You won't need your other windows while you are coding, but keep them open anyway.

Here is what you will see in the coding window:



It

automatically creates the Public Class for the Form and inside this is a subroutine definition for your button. Do not edit this code. You will write your code *inside* the Private Subroutine for BtnAdd. On line 3 above, you can see there is a space to start writing.

Visual Basic requires four sections in a program:

1. Declare variables
2. Read in data from an object to a variable
3. Process the data.
4. Display the data in an object.

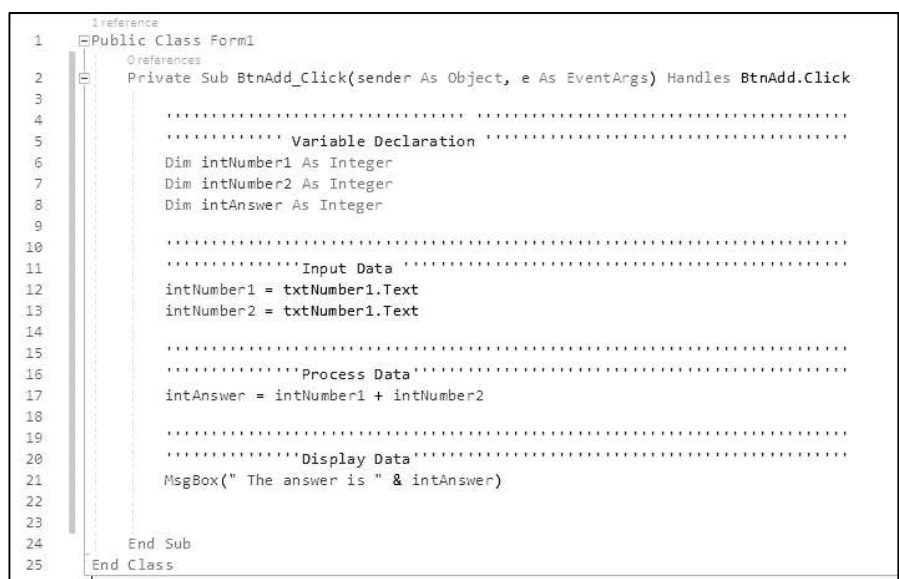
Below is the VB code that will read in two values via the textboxes and add them together. The answer will be displayed in a pop-up message box. You will notice that I have entered a row of single inverted commas. These are 'comments' that are not executed. It allows me to make the code easier to read.

First we declare our variables as integers.

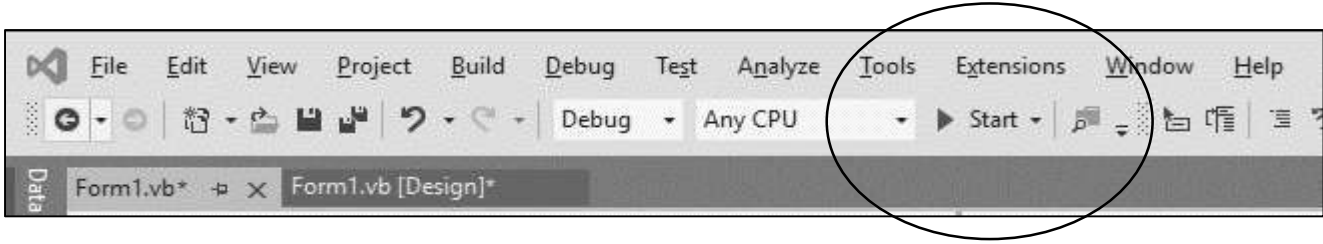
Then we read our data in from the text property of our textbox objects.

We add our two values together for our intAnswer variable.

Finally we display a comment and the data held in the variable intAnswer.

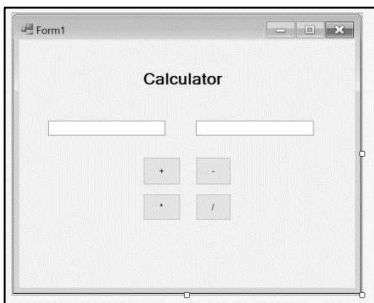


Now all we have to do is run it! Just click the ► Start button and it will run your code!



Now let's develop our calculator further. Add a 'minus' button and a 'multiply' button. The minus is right next to the plus on the keyboard. The character we use for multiplication is * which is shift 8.

Everything should still work fine... BUT if we put a 'divide' button in, we will have problems because we have set our variables to integers. If you divide an integer with an integer you get a decimal number. A more flexible option is to use all decimal numbers. In VB, the best data type to use for decimals (or floating point numbers) is called a Double. See the code below for our completed calculator. I have removed the comments so I can fit the whole program on one page.



```
1 Public Class Form1
2     Private Sub BtnAdd_Click(sender As Object, e As EventArgs) Handles BtnAdd.Click
3         Dim dblNumber1 As Double
4         Dim dblNumber2 As Double
5         Dim dblAnswer As Double
6         dblNumber1 = txtNumber1.Text
7         dblNumber2 = txtNumber1.Text
8         dblAnswer = dblNumber1 + dblNumber2
9         MsgBox(" The answer is " & dblAnswer)
10    End Sub
11
12    Private Sub BtnSubtract_Click(sender As Object, e As EventArgs) Handles btnSubtract.Click
13        Dim dblNumber1 As Double
14        Dim dblNumber2 As Double
15        Dim dblAnswer As Double
16        dblNumber1 = txtNumber1.Text
17        dblNumber2 = txtNumber1.Text
18        dblAnswer = dblNumber1 - dblNumber2
19        MsgBox(" The answer is " & dblAnswer)
20    End Sub
21
22    Private Sub BtnMultiply_Click(sender As Object, e As EventArgs) Handles btnMultiply.Click
23        Dim dblNumber1 As Double
24        Dim dblNumber2 As Double
25        Dim dblAnswer As Double
26        dblNumber1 = txtNumber1.Text
27        dblNumber2 = txtNumber1.Text
28        dblAnswer = dblNumber1 * dblNumber2
29        MsgBox(" The answer is " & dblAnswer)
30    End Sub
31
32    Private Sub BtnDivide_Click(sender As Object, e As EventArgs) Handles btnDivide.Click
33        Dim dblNumber1 As Double
34        Dim dblNumber2 As Double
35        Dim dblAnswer As Double
36        dblNumber1 = txtNumber1.Text
37        dblNumber2 = txtNumber1.Text
38        dblAnswer = dblNumber1 / dblNumber2
39        MsgBox(" The answer is " & dblAnswer)
40    End Sub
41 End Class
```

There is a lot repetition. Why on earth do we have to declare the variable for each button? Well, we don't!

In the revamped code below, you can see we have declared the variables as Public. This means all the subroutines can access the global variables from inside the class.


```

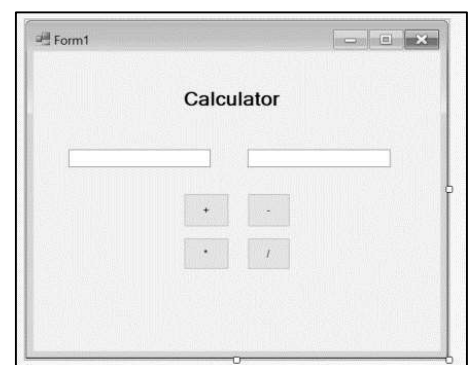
1  Public Class Form1
2      Public dblNumber1 As Double
3      Public dblNumber2 As Double
4      Public dblAnswer As Double
5      Private Sub BtnAdd_Click(sender As Object, e As EventArgs) Handles BtnAdd.Click
6          dblNumber1 = txtNumber1.Text
7          dblNumber2 = txtNumber2.Text
8          dblAnswer = dblNumber1 + dblNumber2
9          MsgBox(" The answer is " & dblAnswer)
10     End Sub
11
12     Private Sub BtnSubtract_Click(sender As Object, e As EventArgs) Handles btnSubtract.Click
13         dblNumber1 = txtNumber1.Text
14         dblNumber2 = txtNumber2.Text
15         dblAnswer = dblNumber1 - dblNumber2
16         MsgBox(" The answer is " & dblAnswer)
17     End Sub
18
19     Private Sub BtnMultiply_Click(sender As Object, e As EventArgs) Handles btnMultiply.Click
20         dblNumber1 = txtNumber1.Text
21         dblNumber2 = txtNumber2.Text
22         dblAnswer = dblNumber1 * dblNumber2
23         MsgBox(" The answer is " & dblAnswer)
24     End Sub
25
26     Private Sub BtnDivide_Click(sender As Object, e As EventArgs) Handles btnDivide.Click
27         dblNumber1 = txtNumber1.Text
28         dblNumber2 = txtNumber2.Text
29         dblAnswer = dblNumber1 / dblNumber2
30         MsgBox(" The answer is " & dblAnswer)
31     End Sub
32 End Class

```

You might have noticed my interface looks nicer now. I put the symbols in the text property of each of the buttons and I have a heading.

The heading is just a label object you can find in your Toolbox. You can format the font type and size in Properties window. Just select font and click on the three dots ...

You can also change the background colour of the Form and other formatting. Just ensure the Form is selected, go to the Properties Window and select "BackColor". There is a pulldown menu and you can choose "Custom" and a wide range of colour options will be available.



Chapter 3

The Age Calculator

We are going to use a very cool built in object we can use in our programming called the Date/Time Picker. You can find it in your Toolbox. We are going to create the interface below. The heading is a Label with "Age Calculator" in the text property. Below it is another label named "lblToday" with 'Today' in the text property. There are labels with 'Name' and 'Date of Birth' in the text property. The input for Name is a textbox called "txtName". The DateTimePicker is called "dtpBirthday". The button is called "btnAge" with 'How old am I?' in the text property and finally a label called "lblAge" with I am... years old" in the text property.



```
1 Public Class Form1
2     Private Sub BtnAge_Click(sender As Object, e As EventArgs) Handles btnAge.Click
3
4         '..... Declaration and Input'.....
5         Dim strName As String = txtName.Text
6         Dim DOB As Date = dtpBirthday.Value
7         Dim Today As Date = Date.Now
8         Dim Age As Integer
9
10        '..... Calculation of Age'.....
11        Age = DateDiff(DateInterval.Year, DOB, Today)
12
13        '..... Output to Screen'.....
14        lblAge.Text = (strName & " , you are " & Age & "years old")
15
16    End Sub
17 End Class
```

We have contained all the variable declarations within the button's subroutine. When the declaration is in the Sub we can read the data in the same line as the Dim statement. Line 5 shows that the Variable "strName" is declared as a String and read in from the textbox "txtName". We can then declare "DOB" as a Date and read it in from our DateTimePicker "dtpBirthday". We have a variable called "Today" which is also a Date data type and it collects today's date from the system clock.

Age is then calculated by finding the Date Difference between "DOB" and "Today". You can see that the Data Interval is using only the Year component of the date (not the month or day). Finally, the "Age" is displayed within a statement in the text property of the "lblAge" object.

Now you might run your program and find it is not correct. Perhaps your birthday is later in the year. We need to test the Month component of the date.