

Visual Basic – Calculations

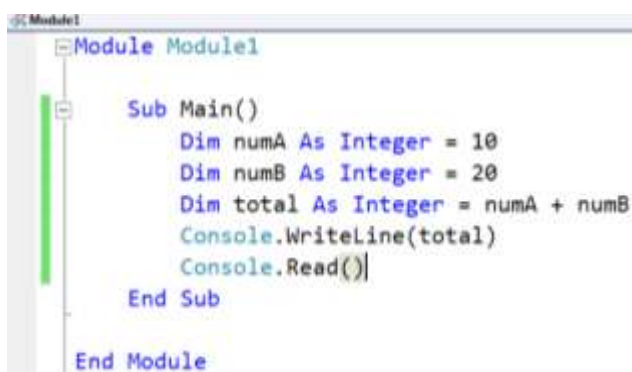
Hello, if you are following us thus far in the tutorials we have covered string in good detail and also shown a little use of numbers or integers in visual basic. Right now we need to learn more about integers because computer love to do calculations. I mean they do it all day every day.

Let's start by opening visual studio and start a new visual basic project in console application.

Call this one calculator.

We are not only going to cover integer in this we will also cover double which allows us to calculate in the decimal numbers. Yippy right.

Write the following in the main function



```
Module Module1
    Sub Main()
        Dim numA As Integer = 10
        Dim numB As Integer = 20
        Dim total As Integer = numA + numB
        Console.WriteLine(total)
        Console.Read()
    End Sub
End Module
```

As you can see we have 3 integer variables in there. numA = 10 and numB = 20 there is also a total variable which is adding both numA and numB together. After that we are showing the total of this using console write line function and simply including total to show the value to the console screen.

Note – we are not using total.toString() this time, why?

Sometimes when you have other strings present in the same line we need to convert numbers to strings in order to reduce the confusion. If all the elements are text then its easier for the program to compile it all together.

Run the program now and it will return 30.

How about we get the user to input both numbers for us eh?

Lets do it.

```

Module Module1
    Sub Main()
        Dim numA As Integer = Console.ReadLine()
        Dim numB As Integer = Console.ReadLine()
        Dim total As Integer = numA + numB
        Console.WriteLine(total)
        Console.Read()
    End Sub
End Module

```

See we changed the numA and numB variable value to console.ReadLine(). This read line function will wait for the user enters a number. Run the program now and it will simple wait for the input. See the console read function in the last line before the end sub we used to display out application is pretty much doing the same thing. We have set the console application it only exits once the user presses something in the end otherwise it will stay running in the memory.

Run the program now.

Lets be a little descriptive to our program you know because we are nice programmers.

```

Console.WriteLine("Enter the first number")
Dim numA As Integer = Console.ReadLine()
Console.WriteLine("Enter the second number")
Dim numB As Integer = Console.ReadLine()
Dim total As Integer = numA + numB
Console.WriteLine("Your Total is: ")
Console.WriteLine(total)
Console.Read()

```

Now run the program and see it helps you by giving you directions to what you need to do.

If want to change it to do substractions do the following

```

Console.WriteLine("Enter the first number")
Dim numA As Integer = Console.ReadLine()
Console.WriteLine("Enter the second number")
Dim numB As Integer = Console.ReadLine()
Dim total As Integer = numA - numB
Console.WriteLine("Your subtraction is: ")
Console.WriteLine(total)
Console.Read()

```

Multiplication

```

Console.WriteLine("Enter the first number")
Dim numA As Integer = Console.ReadLine()
Console.WriteLine("Enter the second number")
Dim numB As Integer = Console.ReadLine()
Dim total As Integer = numA * numB
Console.WriteLine("Your multiply total is: ")
Console.WriteLine(total)
Console.Read()

```

Division

```

Console.WriteLine("Enter the first number")
Dim numA As Integer = Console.ReadLine()
Console.WriteLine("Enter the second number")

```

more tutorials on www.mooict.com

```
Dim numB AsInteger = Console.ReadLine()
Dim total AsInteger = numA / numB
Console.WriteLine("Your division total is: ")
Console.WriteLine(total)
Console.Read()
```

We just hit a problem with division. Lets do this calculation $9/2$ in the division. The answer it returns is 4. Its not wrong and its not right. The answer should be 4.5.

We have to fix it now. If you remember integers to don't support decimal numbers. So we need to use something that does support it.

How about double. Double is a data type that holds numbers and ching ching decimal points.

We can have any decimal points we want.

Change the division code to the following

```
Console.WriteLine("Enter the first number")
Dim numA AsInteger = Console.ReadLine()
Console.WriteLine("Enter the second number")
Dim numB AsInteger = Console.ReadLine()
Dim total AsDouble = numA / numB
Console.WriteLine("Your division total is: ")
Console.WriteLine(total)
Console.Read()
```

Notice the highlighted code. We have changed total from integer to double. Now if you run the program and give the first number 9 and second number 2 it will return 4.5. Which is correct.

We done it yeaay.