Year 12 : Visual Basic Tutorial.

More about … Identifiers : Variables and Constants.

STUDY THIS

Computers process data – that's what they do!

Data is input, then it is processed, then the results are output. The data that is processed may be of a number of different types, but every item of data used by a program must be **declared** – ie. The computer must be told beforehand what data is used, what it is called and what type it is.
This is done using a variety of different statements…

Dim

Eg. If an integer variable is going to be used to store an exam mark, we may use…

```
Dim ExamMark As Integer
```

'ExamMark' is the identifier name; and it is of type Integer.

Const

Eg. If a constant is going to be used to store the VAT percentage rate, we may use …

```
Const VATPercentageRate As Single = 17.5
```

HINTS :

- Always use **self-documenting code** - meaningful names for your identifiers. This will be a good habit to adopt, and will help you develop your programs. (Don't be lazy about typing in long identifier names like 'CustomerFirstName'.)

- Always use a constant if possible. This will make it easier to change the values of the data later. In fact only one change should be made – instead of changing the values all the way through the program!

Local and Global Variables

If a variable is declared inside a **subroutine** then it is only allowed to be used inside that subroutine. This is called a **local variable**. Once the subroutine has been run, the space used to store the variable is released by the computer to be used by other processes.

If a variable is declared inside a **class**, it may be used in **any** of the subroutines inside that class. This is called a **global variable**. The computer reserves space and protects it for the whole time the form is opened.
If you want a global variable (or constant) that can be used throughout **all forms** (classes) of a project use the **Public** declaration…

Eg.

```
Public FilePath As String
Public Const Pi As Double = 3.1415927
```

Operators.

The basic operators that can be used are shown in the table below:

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide
\	Integer division
Mod	The remainder when numbers are divided
^	Exponent (power)
&	String concatenation (joining)

Examples : (Assuming these declarations and values…)

```
'Variable declarations
Dim Num1 As Single, Num2 As Integer

'Assign values to the variables
Num1 = 13
Num2 = 5
```

Then…

Num1 **/** Num2	=	2.6
Num1 \ Num2	=	2
Num1 **Mod** Num2	=	3
Num1 **^** Num2	=	371293

String **concatenation** is the correct word for 'adding' two strings together.

Eg. "TOM" **&** " " **&** "JONES" = "TOM JONES"

(**NB** You can use the operator '**+**' to concatenate strings if you prefer…)

Summary

All **data** used in a program is labelled with an **identifier** - a name that makes it easy for us to recognise.

A **variable** is an identifier that may change each time we run a program.

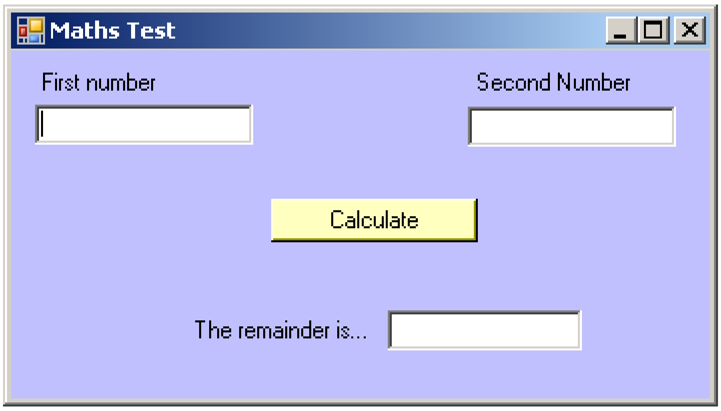
A **constant** is an identifier that is the same every time the program is run.

Visual Basic Challenges 2

HANDS ON

[1] Create a **Factor Test** program that displays the remainder when one number is divided by another number.

The interface should look like the form below…

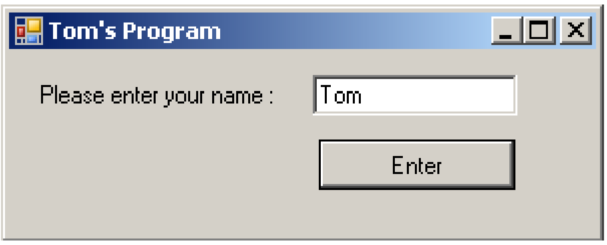


Test Data : 24 divided by 5 has a remainder of 4.
30 divided by 6 has a remainder of 0 (6 is a factor of 30)

Use your program to find the factors of 189
(HINT A factor will give a remainder of 0)

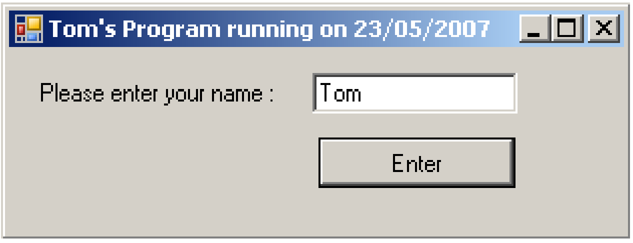
[2] (a) Create an application that allows the user to input their name (Eg. **Tom**), and when an 'Enter' button is clicked, the name of the form at the top changes to '**Tom's Program**'

HINT : When coding the program, the Form is referred to as Me.



RESEARCH NEEDED

(b) Now try adding the current **Date** as well…


Tom's Program running on 23/05/2007
Please enter your name :
Tom
Enter