# Programming

**Learning here:**

Coding techniques:

 Sequence

 If statements

 If … then …. Else

 Events (click, on focus, mouse over, text change

 Input box

 Output box and labels

Design Techniques:

 Colour (color in this program because of American spelling)

 Shape

 Position

 Naming techniques (we use Hungarian)

 How to create code blocks with correct start finish instructions

Screen management:

 Tool box position

 Properties position during design

 Solution explorer, form,

What this will do: (is this part of the description: analysis, design or?)

*Check if an input text is a number*

*Respond if it is*

When you first start the program creates the form for you

You will need an exit button - the only code that it contains is **End**

It needs to be positioned on bottom right approx. (why ? think)

Drag in a button or double click on the button in the common toolbox

Go to the properties pane

Set the name property to btnExit (note the deliberate use of CamelCase) – this name is what the program calls the button

Set the text property to Exit (this is what appears on the face of the button)

Double click on the button

Notice that the code block is created

It has an event as part of the lead in to the block and what it handles at the end of the top line. At the bottom of the block is end sub

Never remove the end class code – it will wreck your program if it is gone

Because this is a button the default event is a click and you have achieved the default event because you just clicked on it.

Note that when you click on it and go to the coding screen you no longer need the toolbox pane and properties pane. However unless you are short of screen real estate (such as on a laptop) just leave them there until you have finished the design /coding phase

Type end inside the code block.

Notice that VB.net immediately starts offering suggestions. Accept a suggestion by arrow to it and click tab or click the mouse on it or … Sometimes the suggestions are very helpful and save typing and reduce mistakes in typing ( a REAL benefit!) VB never offers a suggestion that is NOT syntactically correct.

Now click back to the design tab ( at top of that part of the window)

 **Did you save yet? …**

Add a text box

Set the name property to txtIn (- short for input)

Set the multiline property to true (not really necessary here but later …)

Multiline allows you to resize the box. You can set this just by double clicking in the property value part of the property pane

Add a label

Place it above the text box

Set the text property to Enter number here

There is no need to set the name of the label as it will never be addressed by the running program

Add another label

Position it about the middle of your form

Set the name property to lblGood (lbl for label)

Set the text property to Well Done

Set the autosize property to False

Set the back color to a custom color something like Aqua

Set the visible property to False - note that the label is still there !! (This is the design pane)

Click the small plus sign in front of the Font property +

This causes the font properties to expand

Set the font size to 24

Set the fore color to say **mediumblue** you can access Named colors using the web tab in the color property

Set the textalign property to Middle center

Add another label

Position it about the middle of your form partially overlapping the previous label

Set the name property to lblBad (lbl for label)

Set the text property to That is not a number

Set the textalign property to Middle center

Set the autosize property to False

Set the back color to a custom color something like Coral

Set the visible property to False - note that the label is still there

Click the small plus sign in front of the Font property +

Set the font size to 24

Leave the fore color as the control color – normally it is black

Add a button

Position it to the right of the input box

Set the name property to btnCheck

Set the text property to Check My Number

Double click on btnCheck

Add this code: (the bits in bold only and TYPE it in YOURSELF)

Dim stuffIn as string

 (everything except stuffIn should go blue to indicate that it is reserved words)

 Dim indicates that stuffIn is a variable which will be used later in the program

 String indicates that it will be a string of characters

Many people also put varStuffIn as that indicates it is a variable

I often do not because there are not that many items in our programs but it is better practice

Dim num as integer

 Num is not a reserved word and the name itself indicates that it is a number

stuffIn = txtIn.text

 this imports the stuff in the text box into the variable

Do not do this section:

*num=val(stuffIn)*

 *this converts the text to a Value – if the text is not a number then it converts it to zero*

The val( ) method and the str( ) method are from much older versions of VB.Net and generally still work but do not handle problems terribly well …

So I have researched another way ( gotta love google)

if int32.tryparse(stuffin, num) then

lblGood.visible = true

else

lblBad.visible = true

The **end if** should already be there

*TryParse ( ) works by checking if it can be converted to an integer and if it can then sticks the value in num and returns a Boolean value of true. If it can’t be converted then it returns a Boolean value of false and num is left blank*

*This also works with double.tryparse ( ) so can be very handy.*

*It is not necessary to know for the exam*…..!!!!!

Now up the top is a small green triangle (play button) Use it.

Enter a number; click the check button

Enter a letter; click the check button

Think about it…

In the Design pane

Add a button and place it just under the check my number button

Name it btnClear

Text on button : Clear the Number Box

Double click on the button

Add this code:

lblBad.visible = false

lblGood.visible = false

txtIn.text= “”

the double quotes contain nothing so that is what is put in the text box

Another way is to type:

txtIn.clear

instead of = “” -- technically more pure!

Now try it