CT240: Programming

- Date:
  - 09-September 2011
- Topic:
  - VB.NET Introduction
  - Labs will start next Thursday … will discuss lab sheet at lecture on Tuesday

VB.NET Overview

- For rapid and relatively simple development of full Windows applications.
- Provides graphical building blocks from which you build your Graphical User Interface (GUI)
- BASIC programming code tied to elements of the GUI.
- VB programs can communicate with other Windows applications.
- Other .Net languages: C++, C#

VB.NET Overview

2 features make VB different from traditional programming tools:

- You literally draw the GUI using *objects*
- Command buttons, text boxes and other objects that you've placed in a blank window will automatically recognise user actions such as mouse movements and button clicks.

Visual Basic Vs Conventional Languages

Programs in conventional languages run from the top down. Execution starts from the first line, and moves with the flow of the program to different parts, as needed.

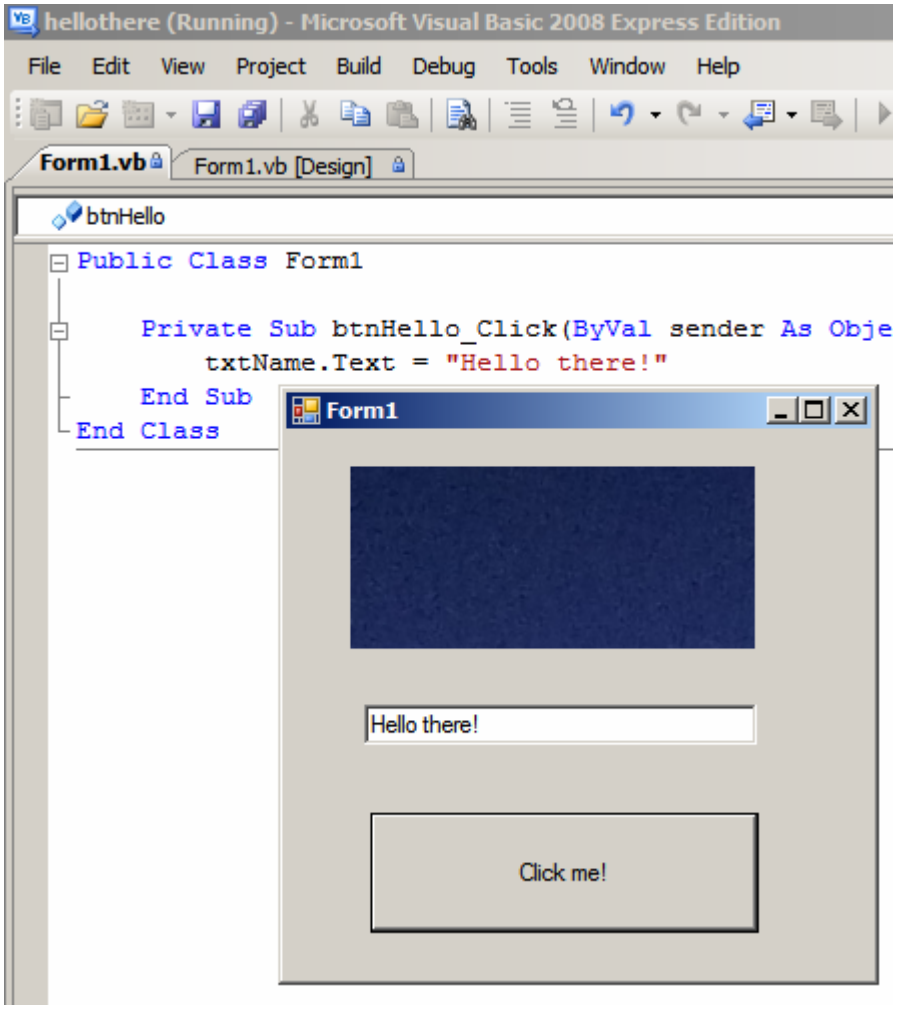
Objects in VB recognise *events*; how the objects respond to events depends on the code you write. You always need to write instructions in order to make controls respond to events.

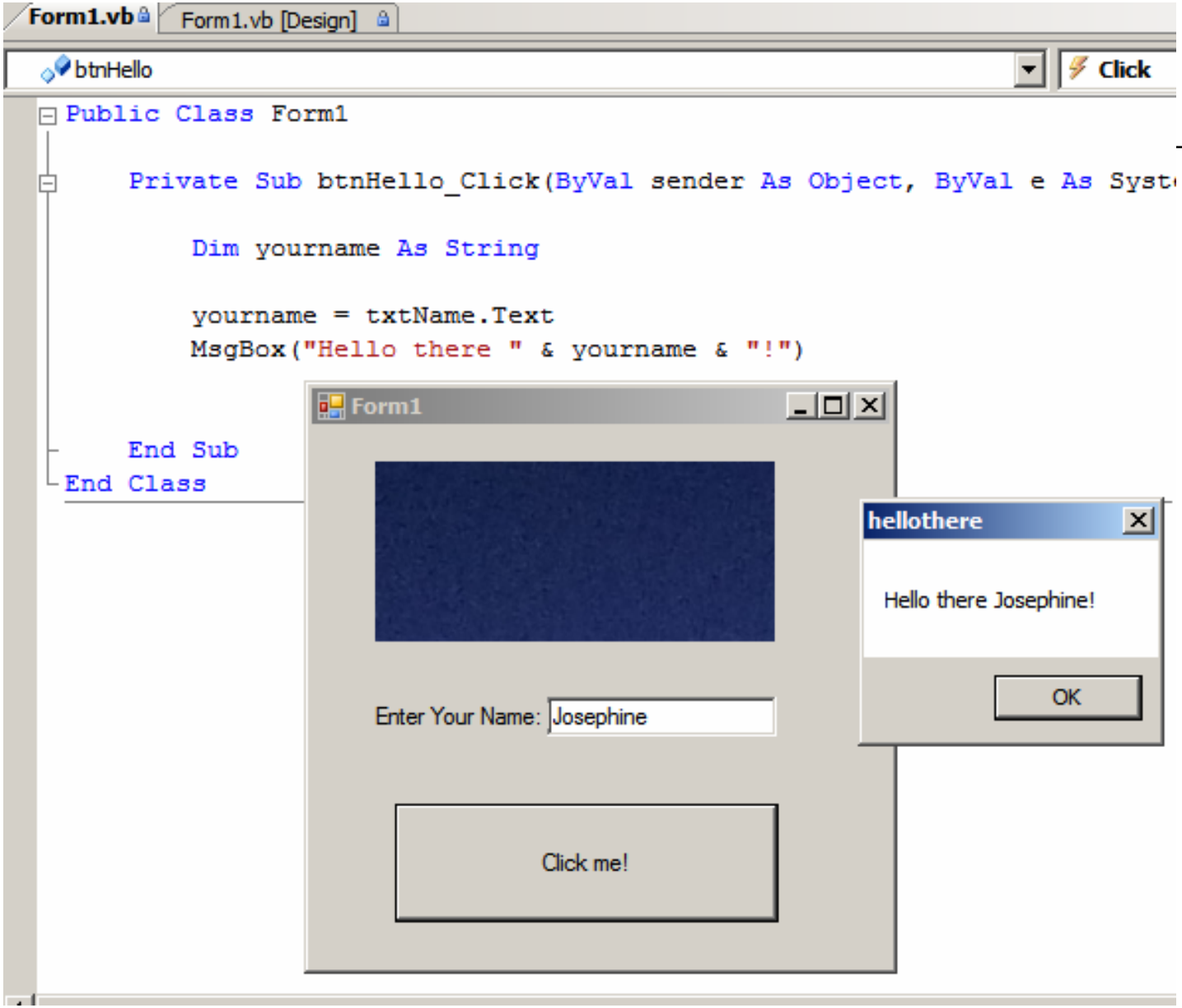
Example from last year in Python

```
IDLE 2.6.6
>>> def SayHello():
        print "Hello"
        print "there"


>>> SayHello()
Hello
there
>>> def SayHello(name):
        print "Hello there", name


>>> SayHello("John")
Hello there John
>>>
```




Form1.vb
Form1.vb [Design]
btnHello
Click
Public Class Form1
Private Sub btnHello_Click(ByVal sender As Object, ByVal e As Syst
Dim yourname As String
yourname = txtName.Text
MsgBox("Hello there " & yourname & "!")
End Sub
End Class
Form1
Enter Your Name: Josephine
Click me!
hellothere
Hello there Josephine!
OK

Application Development in VB.NET

3 steps:

1. **Create the interface** – drag and drop, position and size the objects.
2. **Set properties** - relevant properties for the objects-especially `Name` and `Text`
3. **Write the code** that executes when the events occur.

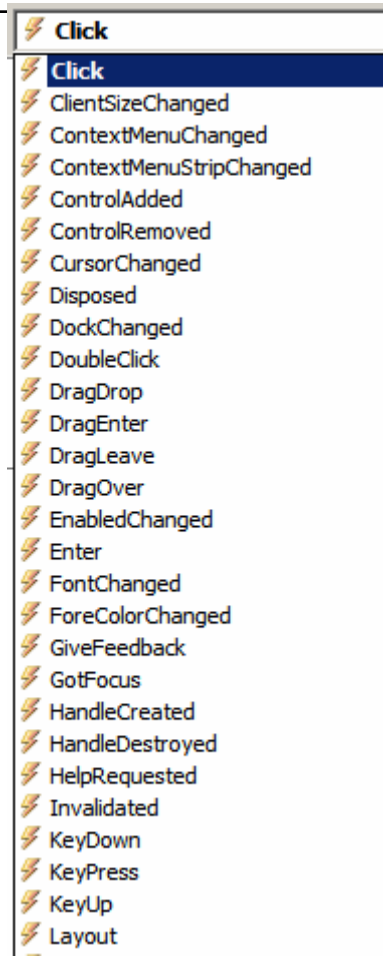
When the program is running

1. VB monitors the objects in the window to detect any event that an object can recognise.
2. When VB detects an event, it examines the program for a corresponding event procedure.
3. If you have written an event procedure, VB executes the instructions and goes back to step 1.
4. If you have not written an event procedure, VB goes back to step 1 and waits for the next event.

This continues until the application ends.

Event-Driven Programming

- An *event* is an action recognised by a form (window) or control.
- *Event-driven* applications execute modules of code in response to events, often from the user.
- In VB, each graphical control has a predefined set of events, e.g. `Click, GotFocus`
- Each *event* can have associated *code*, which you must write.



Creating a VB program

3 steps:

1. **Create the interface** – drag and drop, position and size the objects.
2. **Set properties** - relevant properties for the objects-esp. `Name` and `Text`
3. **Write the code** that executes when the events occur.

“Hello there!” Example in VB.NET

- Inputs: None
- Process: Display “Hello there” in a text box when a command button is clicked
- Outputs: “Hello there”

Steps:

1. Create the interface:

Use 2 objects a text box and (command) button

2. Set properties:

- text box, with `name` txtName
- button:
  - `name` btnHello and
  - `text` ‘Click Me!’

“Hello there!” Example in VB.NET

3. **Write the code** that executes when the events occur:

for `btnHello_Click`

```
txtName.Text = “Hello there!”
```

“Hello there Josephine” Example

1. Create the interface:

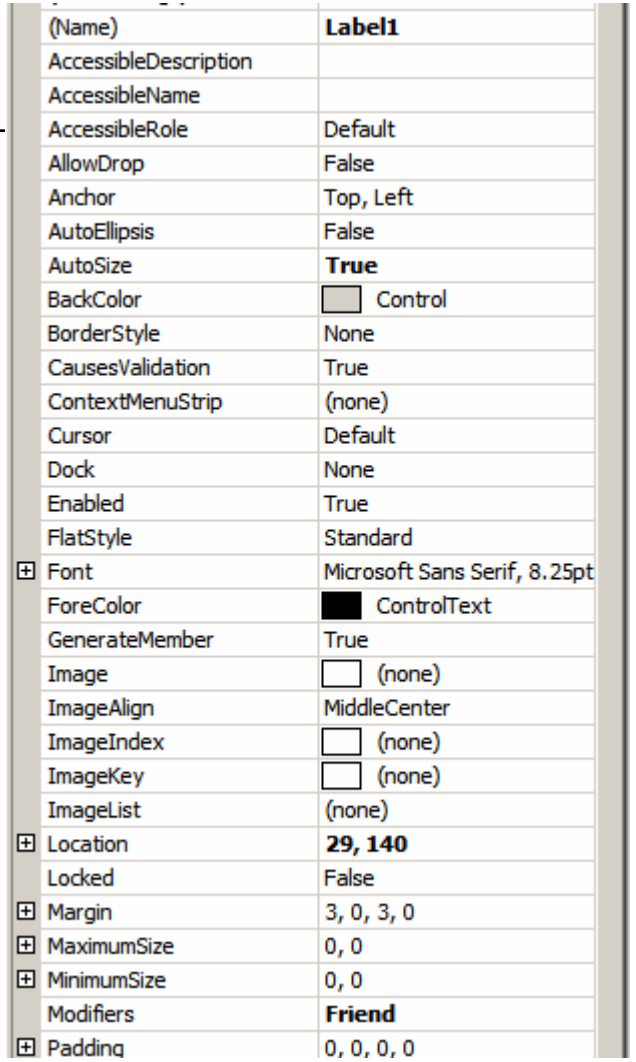
Use 3 objects: a text box and a button and a label

2. Set properties:

- text box, with `name` txtName
- (command) button:
  - `name` btnHello and
  - `text` ‘Click Me!’
- label with `text` ’Enter Name:’

Questions already!?

- Why no `name` property for label object?
- Why no `text` property for text box object?
- There are many properties that can be set for each object – only set the ones you need



(Name)	**Label1**
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
AutoEllipsis	False
AutoSize	**True**
BackColor	Control
BorderStyle	None
CausesValidation	True
ContextMenuStrip	(none)
Cursor	Default
Dock	None
Enabled	True
FlatStyle	Standard
⊞ Font	Microsoft Sans Serif, 8.25pt
ForeColor	ControlText
GenerateMember	True
Image	(none)
ImageAlign	MiddleCenter
ImageIndex	(none)
ImageKey	(none)
ImageList	(none)
⊞ Location	**29, 140**
Locked	False
⊞ Margin	3, 0, 3, 0
⊞ MaximumSize	0, 0
⊞ MinimumSize	0, 0
Modifiers	**Friend**
⊞ Padding	0, 0, 0, 0

3. **Write the code** that executes when the events occur:

for `btnHello_Click`

```
Dim yourname As String

yourname = txtName.Text
MsgBox("Hello there " & yourname & "!")
```

Programming Abstractions

There are 3 main abstractions in the majority of programming languages:

1. **Data** - *instructions operate on data*
2. **Control Structures** - *order of execution of instructions and*
3. **Modules / Sub-programs / Functions**

Data

- **1.1. simple data types** - integers, reals, Booleans, strings, etc.

- **1.2. collection of data / data structures / structured data types** - arrays, stacks, queues, linked lists, etc.

  A few structured data types will be built into programming languages but mostly programming languages provide programmer with constructs to define own data structures.

Control

2. Control Structures - *order of execution of instructions and program*

2.1 statements

2.2 precedence rules

2.3 selection statements

2.4 iterative statements

2.5 recursive statements

Modules/Sub-programs/Functions

3. Modules/Sub-programs/Functions/Objects

- may not always be supported by a programming language

More on Data

Constants

- Value of constant does not change during run-time of program

Variables

- memory location assigned to a particular data object can contain different values during the run-time of the program.
- identifier refers to the name of a variable

Declaring variables …

The system needs to be able to create enough storage space in memory for the data that you are using

- VB.NET must be told in advance the **type** of the data
- This differs to Python
- Also will provide the **identifier name** when declare type

Variable Declarations in VB.Net

Example:

Dim firstnum As Integer

Declares variable *firstnum* as type Integer

Simple Data Types in VB.Net

- Short (16 bits)
- Integer (32 bits)
- Long (64 bits)
- Decimal
- Single, Double
- Date
- Boolean
- Char

Example: Declare 2 integers, named x and y

```
Dim x as Integer
Dim y as Integer
```

Alternatively, can declaring multiple identifiers of the same type on one line

```
Dim x, y  As Integer
```

String Data Type

- A string constant is a sequence of characters that is treated as a single item:

  “This is a string.”

  “Hello there!.”

- String constants must be surrounded by (double) quotation marks.
- A string variable is declared using a statement of the form:

  Dim *variableName* As String

Other Data Types in VB.Net

- Array
- Object

Constants

- Const MAX_CAPACITY As Integer 40

Initialising Variables ….

All variables used in a program/algorithm **must** get a value at some stage. The value may come from:

- the user via text box etc.
- a file
- from or to a function (being passed)
- an assignment statement
- some calculation

Assignment Statement

Format in Visual Basic:

LHS = RHS

where:

- = is assignment operator
- LHS is a variable/memory location
- RHS is constant; variable or expression.
- The RHS is evaluated and assigned to the memory location indicated by the LHS.
- Once the variable is assigned its new value any old value is lost

Example

```
Dim x as Integer
Dim y as integer
x = 1
y = 34
x = x + y
```

- Hence:

  1 = x

  does not make sense. 1 is not a memory location.

Initialising Strings

e.g. to assign the string constant “Ann" to the variable *name*

```
Dim Name As String
Name = “Ann"
```

What data types, names and initial values would you give these things?

- Customer name
- Customer address
- ID number
- Weight of bin
- Balance due
- Price per kg
- Payment due date

Spot the errors …

```
Dim name as Integer
Dim x as Integer

name = “Josephine”
x = “3”
```

```
Dim name as String
Dim x as Integer

name = “Josephine”
x = y + 3
```

```
Dim x, y, z as Integer

3 = x
4 = y
5 = z
```

```
Dim x, y, z as Integer

x = 3
y = 4
z = 3/4
```

Important Points:

- In VB.NET have Objects with associated:
  - Properties
  - Code
- Data declarations with Dim
- Data assignment with LHS = RHS