

1. Introduction to programming

Fundamentals of Computer Science

Engineering Degree – 2010-2011

Ismael Etxeberria Agiriano

20/09/2010



Escuela Universitaria
de Ingeniería
Vitoria-Gasteiz

Ingeniaritzako
Unibertsitate Eskola
Vitoria-Gasteiz

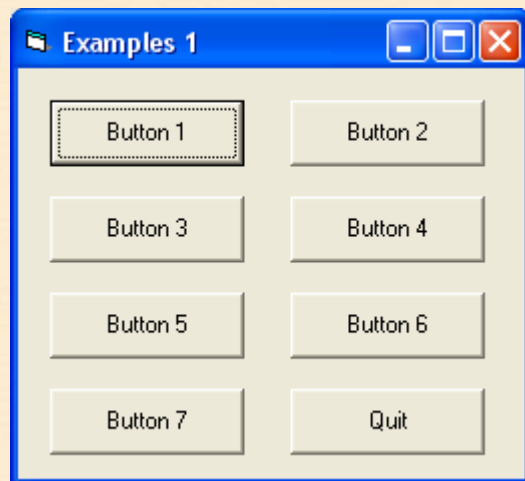


Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Index

1. Introduction to programming

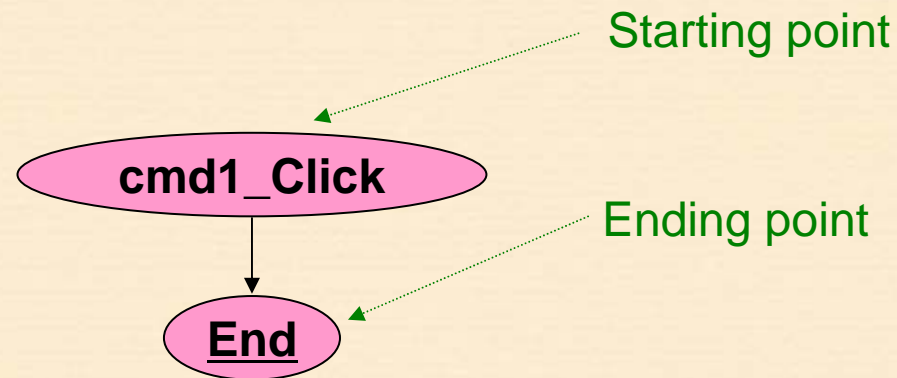


1. Ex01: Empty
2. Ex02: Greeting
3. Ex03: Assignment
4. Ex04: Reading
5. Ex05: Concatenation
6. Ex06: Conversion
7. Summary
8. FC vs VB notation
9. Exercise

1. Example 01

- **Title**
 - Empty subprogram (Button 1)
- **Name**
 - cmd1_Click
- **Description**
 - Subprogram not doing anything
- **Observations**
 - Beginning and ending of a subprogram
 - VB implementation
 - Basic pattern

Ex01: Flowchart



- **Starting point**
 - Subprogram name
 - Control name (cmd1) + event (Click)
- **Ending point**
 - Unique for each flowchart
 - FC code: End

Ex01: VB implementation

```
Private Sub cmd1_Click()  
End Sub
```

Cmd1_Click

End

It automatically sets Private. It is not necessary but it may stay

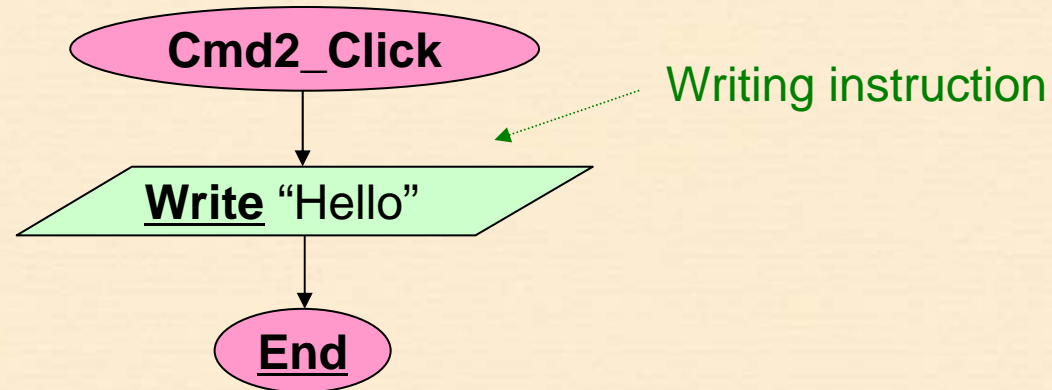
The VB environment proposes the skeleton for the associated subprogram by clicking on the cmd1 command button in design mode

2. Example 02

- **Title**
 - Greeting (Button 2)
- **Name**
 - cmd2_Click
- **Description**
 - Subprogram to say hello
- **Observations**
 - Writing a literal text on the screen



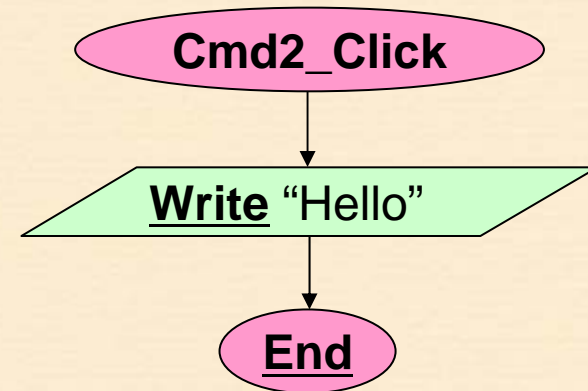
Ex02: Flowchart



- The writing instruction will be expressed in the flowchart as **Write**
- In general we shall not provide details on how to write things in the flowchart. We may have some annotations next to it with that purpose.

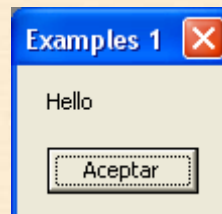
Ex02: VB implementation

```
Sub cmd2_Click()  
    MsgBox "Hello" ' Write  
End Sub
```



A single quote starts a comment:
any text will be ignored

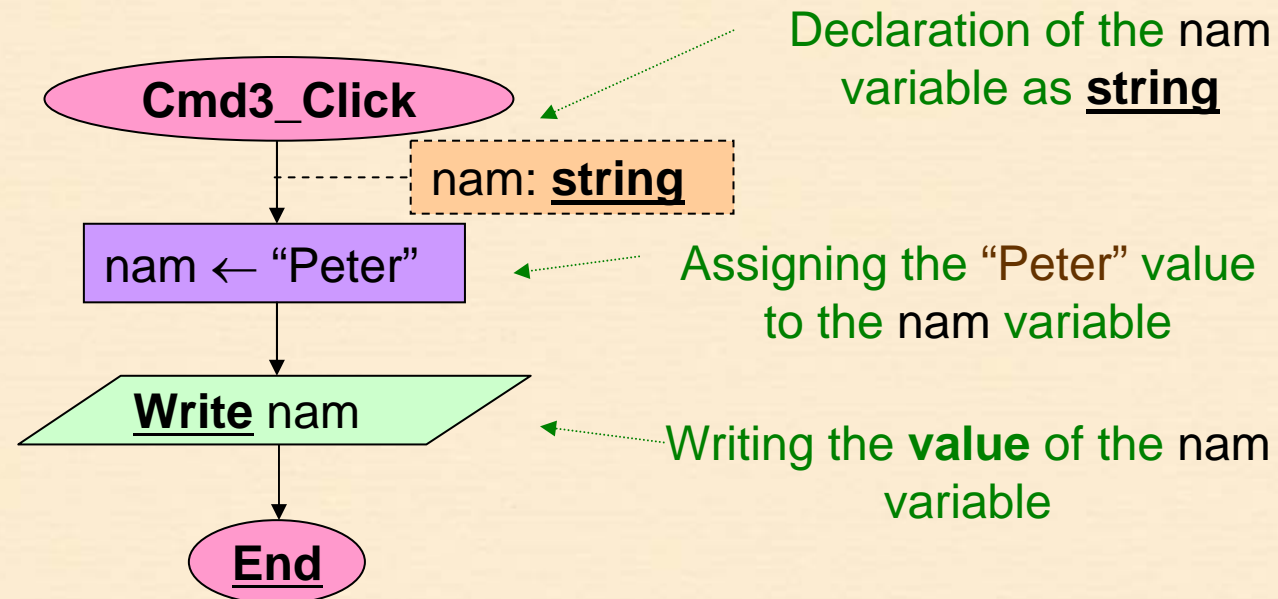
- In VB we utilize the **MsgBox** instruction for writing



3. Example 03

- **Title**
 - Assignment (Button 3)
- **Name**
 - cmd3_Click
- **Description**
 - Definition of a string variable and assignment of a fixed value, displaying it on the screen
- **Observations**
 - Variables declaration
 - Assignment of a value to a variable (\leftarrow)
 - Writing the value of a variable

Ex03: Flowchart



- We must declare all the variables used in the program, indicating their type, for example, **string**
- We underline these keywords (e.g. **string**, **Write**) to differentiate them from the invented names (e.g. **nam**)
- With an assignment a variable receives (**←**) a value

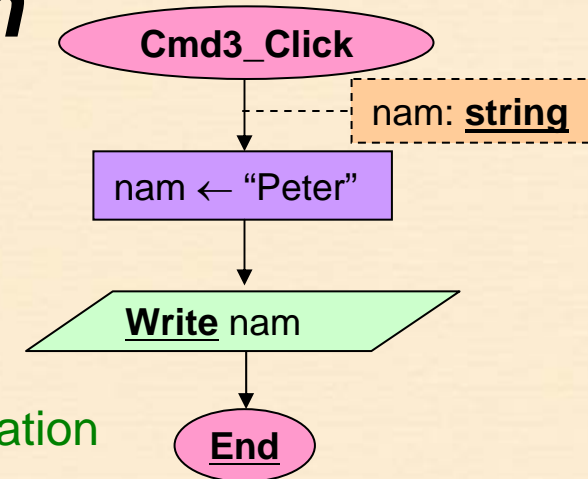
Ex03: VB implementation

```

Sub Cmd3_Click()
  Dim nam As String
  nam = "Peter"
  MsgBox nam
End Sub

```

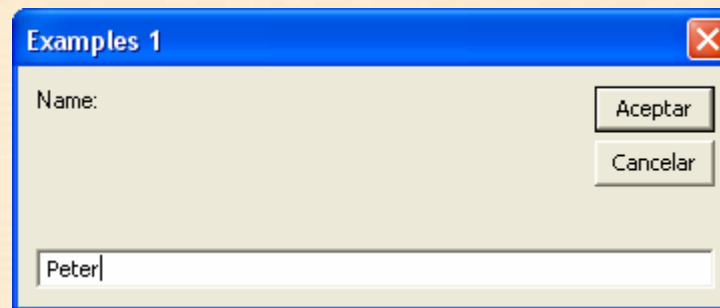
Declaration
 Assignment/Initialization
 Display the value



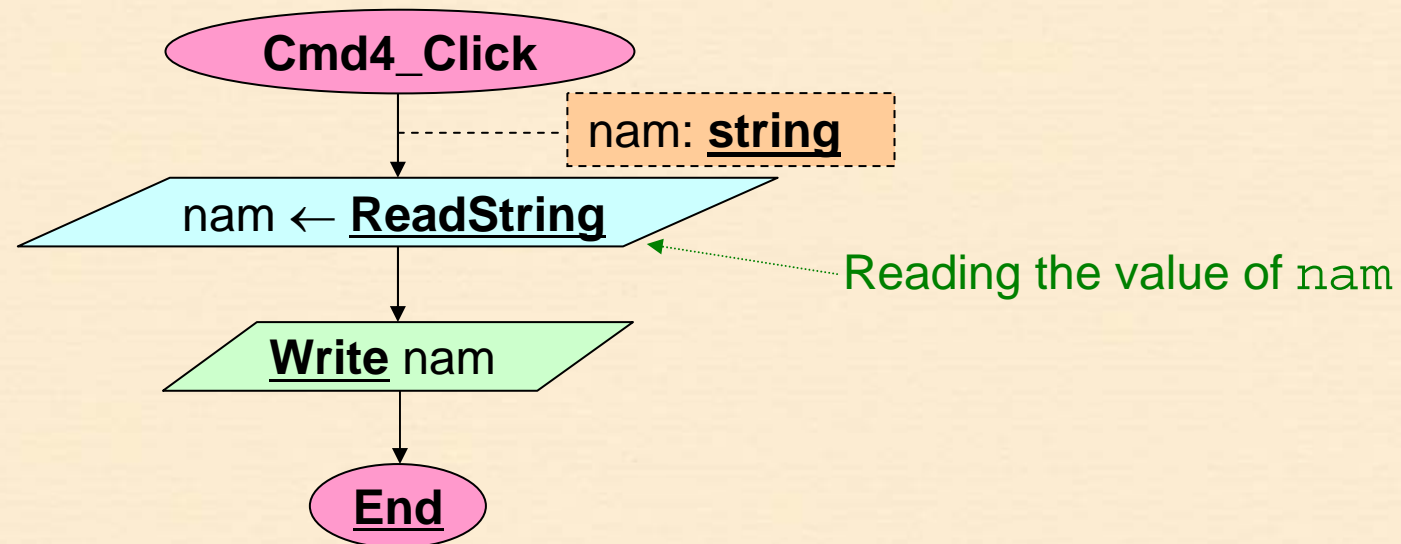
- To declare a variable in VB, after the Dim keyword we specify the **name** of the variable, for example, `nam`, after the As keyword, followed by the **type**, e.g. String
- Assignments in VB are expressed by means of the = symbol
- The left part of an assignment must always have a variable and the right part an expression to be evaluated
- Do not mix assignment and equality!

4. Example 04

- **Title**
 - Reading (Button 4)
- **Name**
 - cmd4_Click
- **Description**
 - Subprogram to read a name and display it on the screen
- **Observations**
 - Reading a value



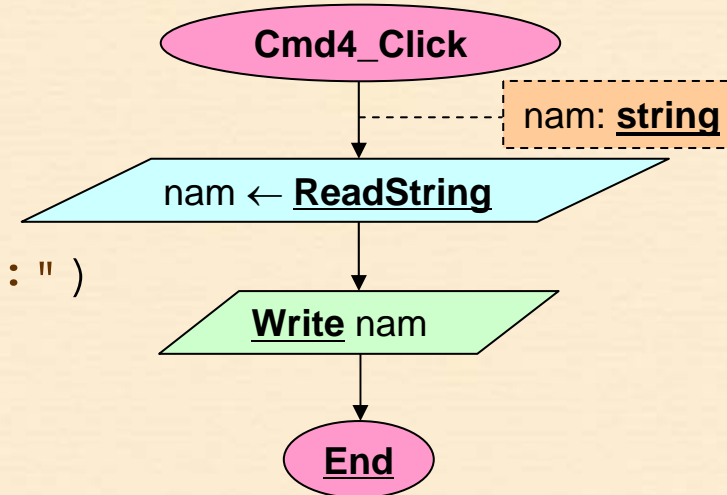
Ex04: Flowchart



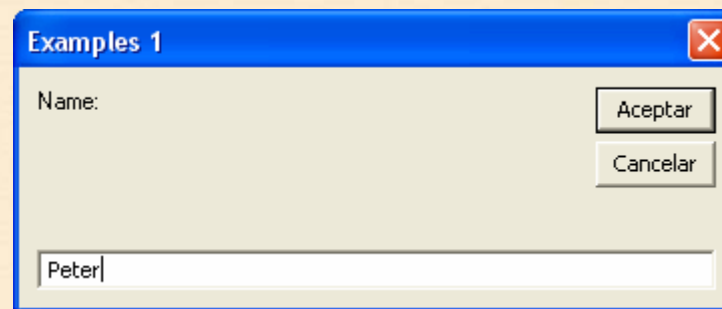
- Instead of assigning a fixed constant value to `nam`, as in Ex03, we now read this value from the keyboard.
- The reading instruction of a string will be expressed as **ReadString** in the flowchart
- After we write the read value on the screen
- Note that we don't give details on how to write things

Ex04: VB implementation

```
Sub Cmd4_Click()  
  Dim nam As String  
  nam = InputBox ("Name: ")  
  MsgBox nam  
End Sub
```



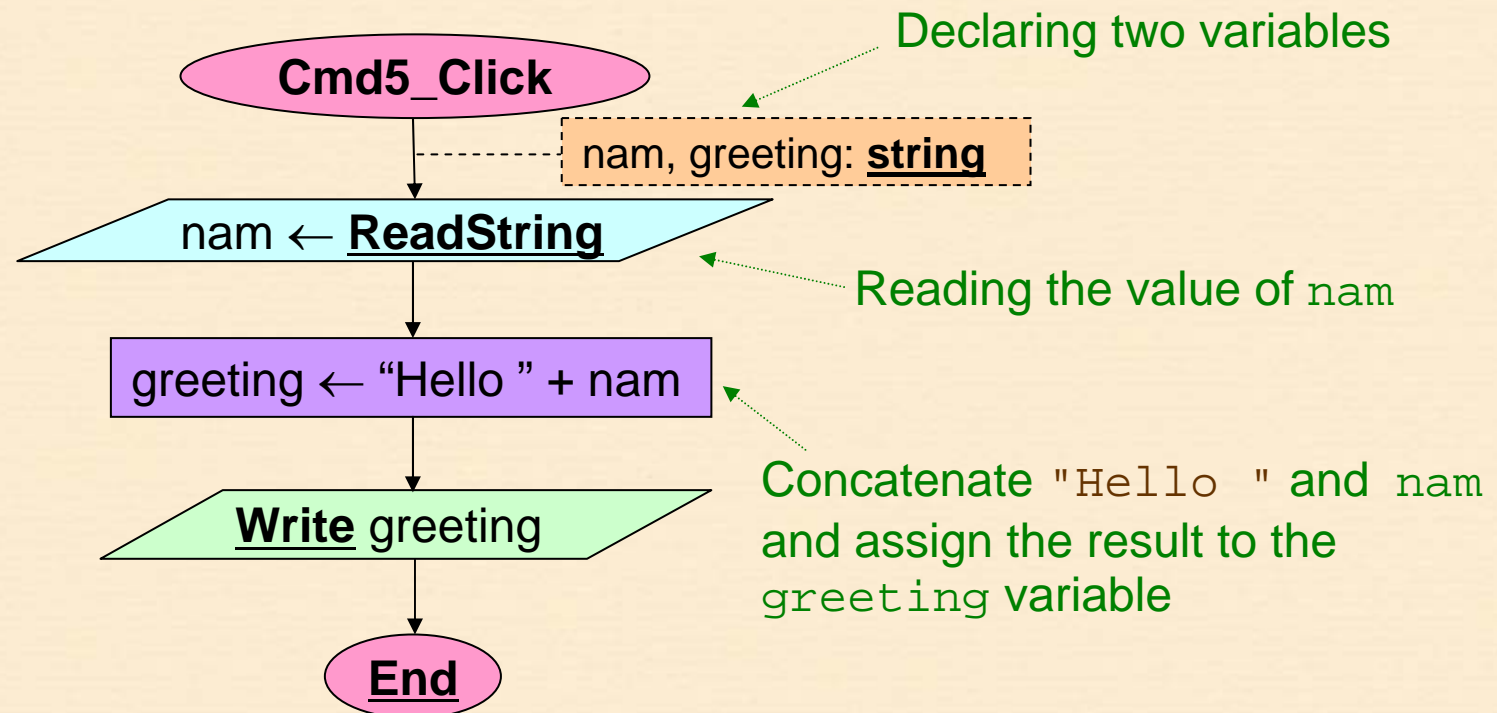
- To read in VB we utilize the **InputBox** instruction, which always returns a string



5. Example 05

- **Title**
 - Concatenation (Button 5)
- **Name**
 - cmd5_Click
- **Description**
 - Subprogram to read a name and obtain a greeting string by concatenating a literal string to it
- **Observations**
 - Multiple variables declaration
 - Expression: concatenation

Ex05: Flowchart



Ex05: VB implementation

```

Sub cmd5_Click()
  Dim nam As String
  Dim greeting As String
  nam = InputBox("Name:")
  greeting = "Hello " & nam
  MsgBox greeting
End Sub

```

Concatenate "Hello " and nam and assign the result to the greeting variable

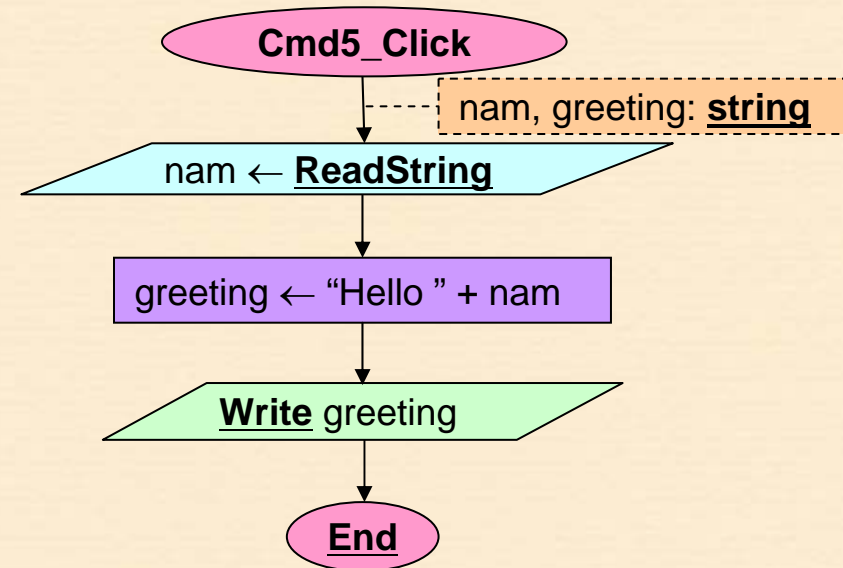
Multiple declaration

```

Dim nam As String, greeting As String

```

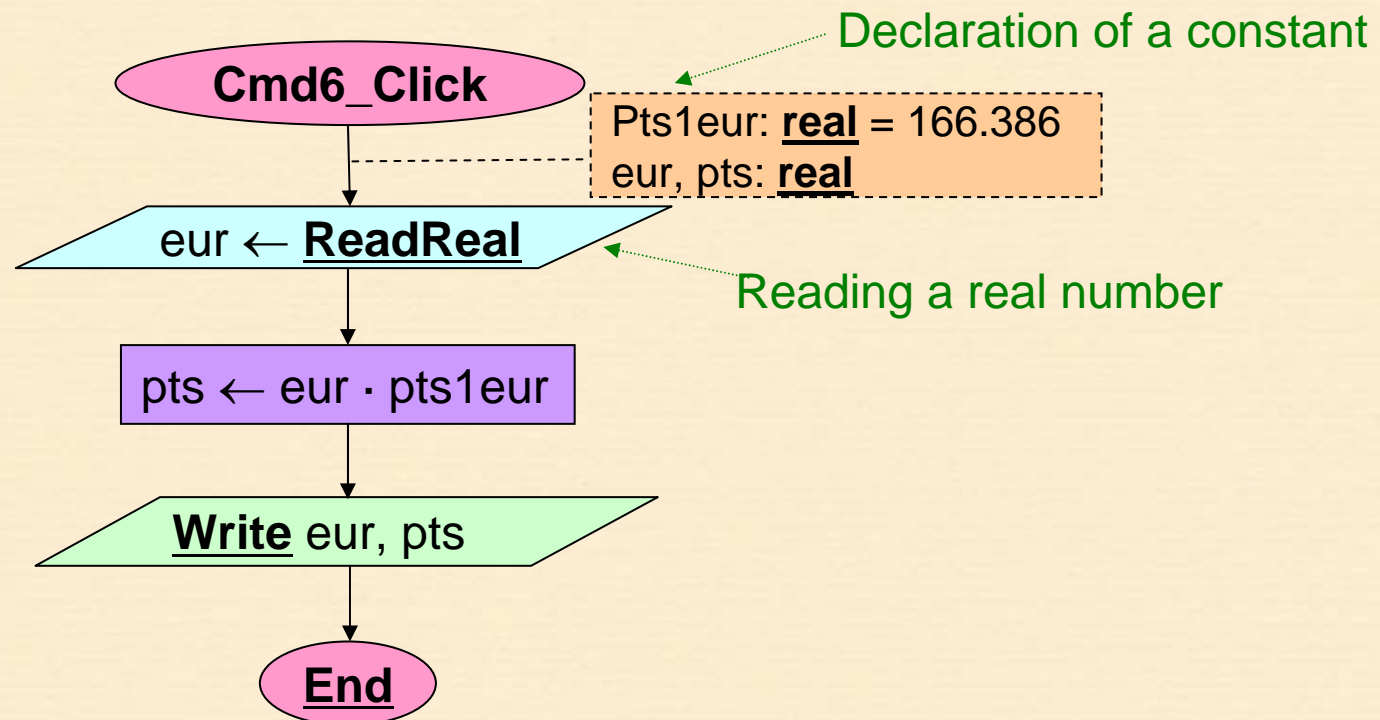
We may declare several VB variables separated by a comma but we must specify the type again



6. Example 06

- **Title**
 - Conversion (Button 6)
- **Name**
 - cmd6_Click
- **Description**
 - Ask for a quantity in euros, convert it into pesetas and display the resulting value
- **Observations**
 - Reading real numbers
 - Expression: multiplying
 - Constants

Ex06: Flowchart



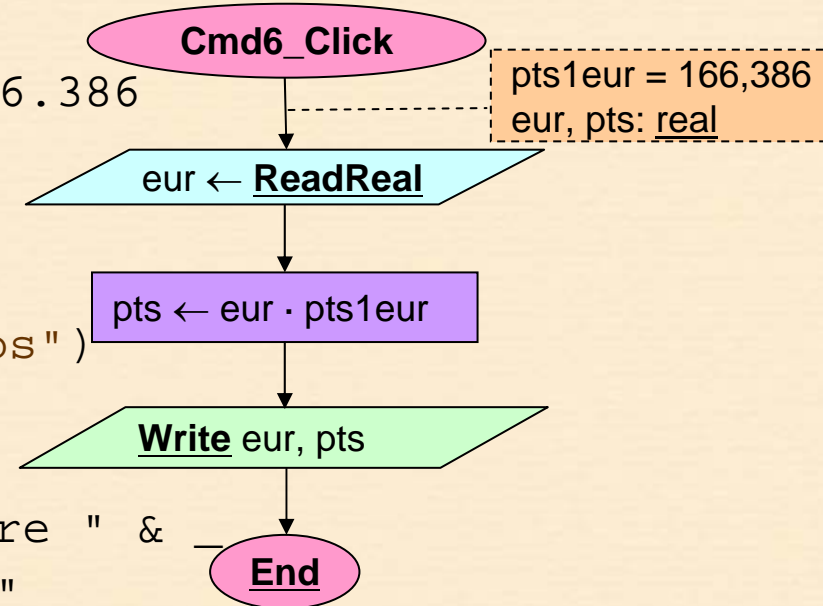
Ex06: VB implementation

```

Sub cmd6_Click()
  Const pts1Eur As Double = 166.386
  Dim eur As Double
  Dim pts As Integer
  Dim s As String
  s = InputBox("Quantity in euros")
  eur = Cdbl(s)
  pts = eur * pts1Eur
  MsgBox CStr(eur) & " euros are " &
         CStr(pts) & " pesetas"
End Sub

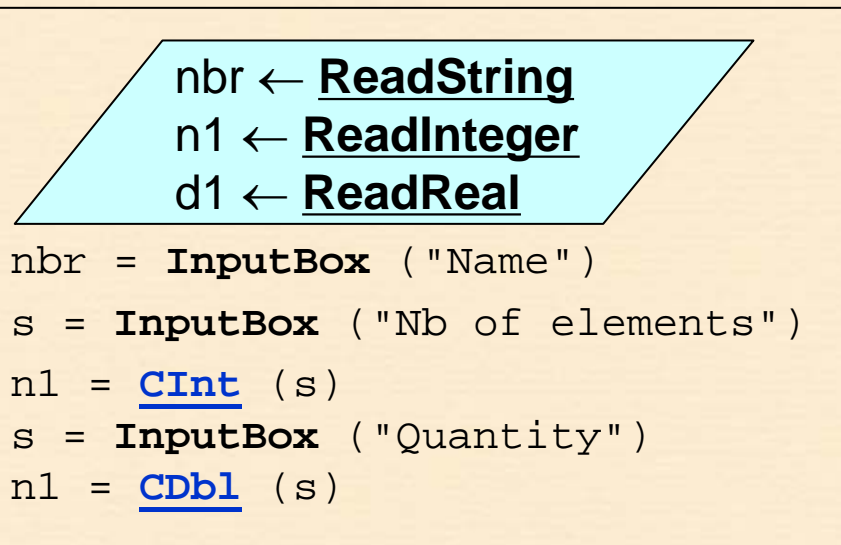
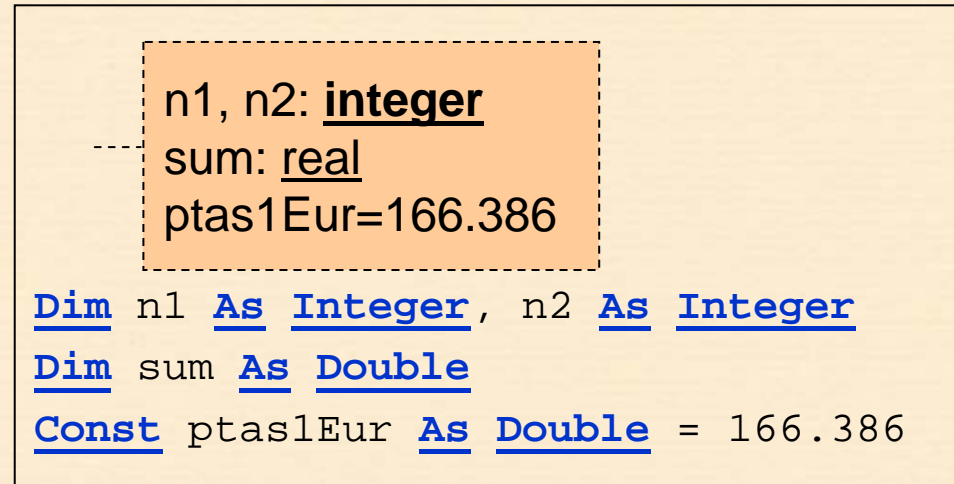
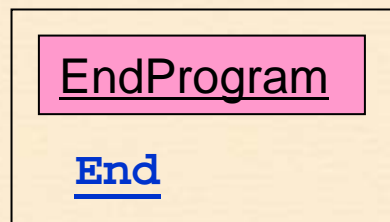
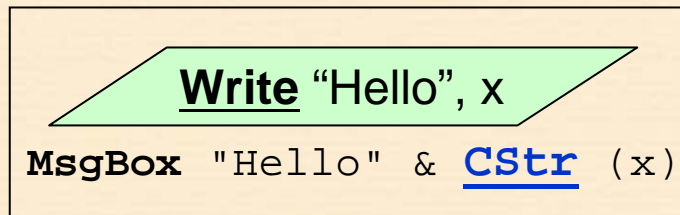
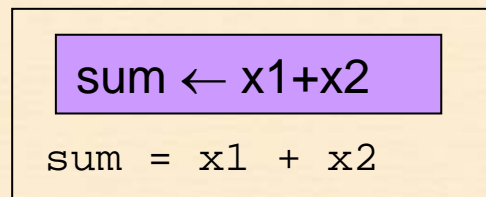
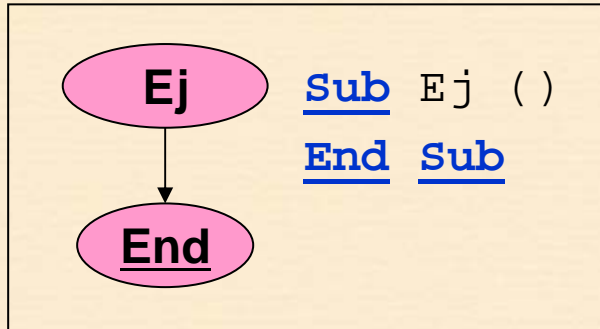
```

Not necessary to specify the type



We use the underline symbol to indicate that the instruction continues on the following line

7. Summary



8. FC vs VB notation (I)

Flowchart	Visual Basic	Comment
←	=	Assignment
+	+	Sum
-	-	Subtraction, change sign
.	*	Product
<u>Div</u>	\	Integer division
<u>Mod</u>	<u>Mod</u>	Division modulus (rest)
/	/	Real division
+	&	Concatenation
<u>integer</u>	<u>Integer</u>	Integer (2 bytes)
<u>real</u>	<u>Double</u>	Real (double precision)
<u>string</u>	<u>String</u>	Character string

8. FC vs VB notation (II)

Flowchart	Visual Basic	Comment
<u>ReadString</u>	InputBox	String reading
<u>ReadInteger</u>	InputBox, CInt	Integer reading
<u>ReadReal</u>	InputBox, Cdbl	Real number reading
<u>EndProgram</u>	End	Finish the whole program exec.
<u>toInteger</u>	CInt	Convert into integer
<u>toReal</u>	Cdbl	Convert into real (double)
<u>toString</u>	Cstr	Convert into string
<u>NewLine</u>	vbCrLf	New line
<u>Write</u>	MsgBox	Write a message box
	Option Explicit	Force variable declarations



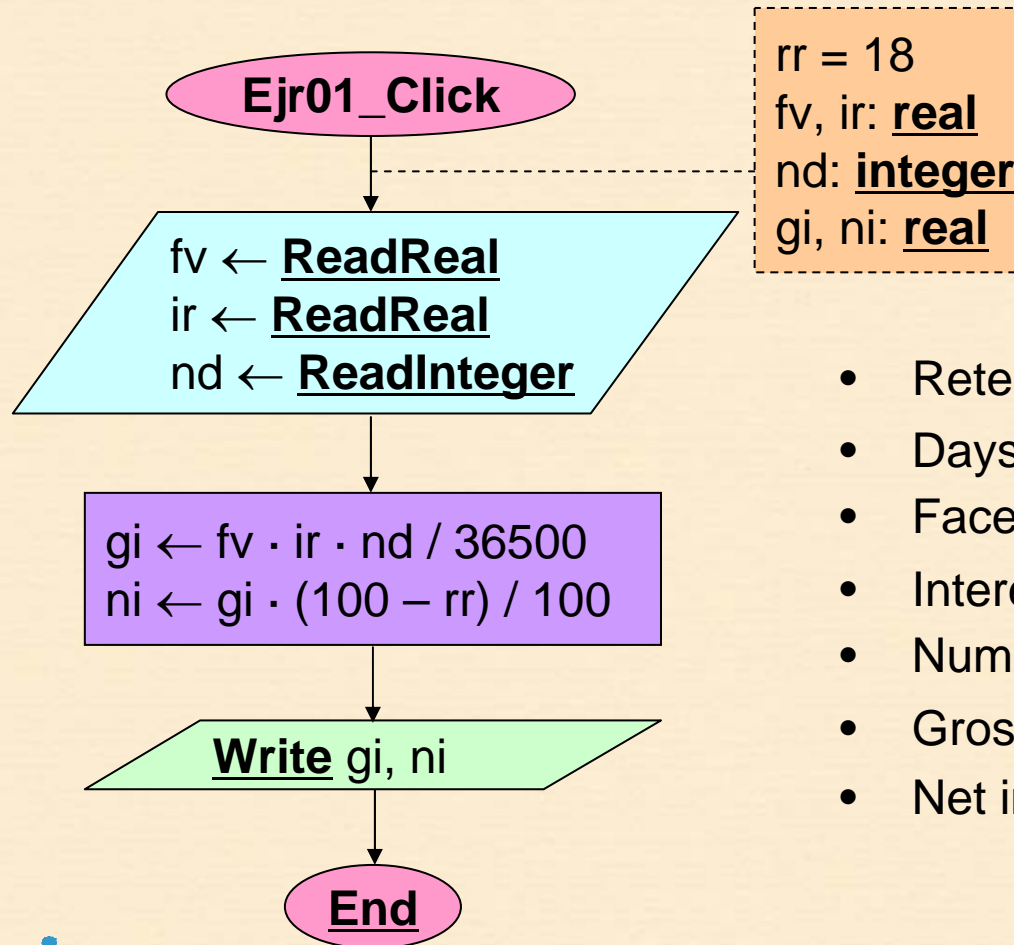
9. Exercise

- **Title**
 - Interests (Button 7)
- **Name**
 - Exr01_Click
- **Description**
 - Design an implement a VB program to calculate the **interest** of a given **quantity** with a yearly basis interest **rate** given a certain number of **days** assuming the year has exactly 365 days. Obtain the **gross interest** and the **net interest** knowing the we suffer a retention of an 18%.

Exr01: Analysis

- Known information
 - Days in a year = 365
 - Retention rate = 18% (rr)
- Information to read
 - Face value (f_v)
 - Interest rate (i_r)
 - Number of days (n_d)
- Information to calculate
 - Gross interest (gi)
 - Net interest (ni)

Exr01: Flowchart



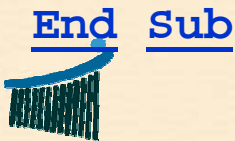
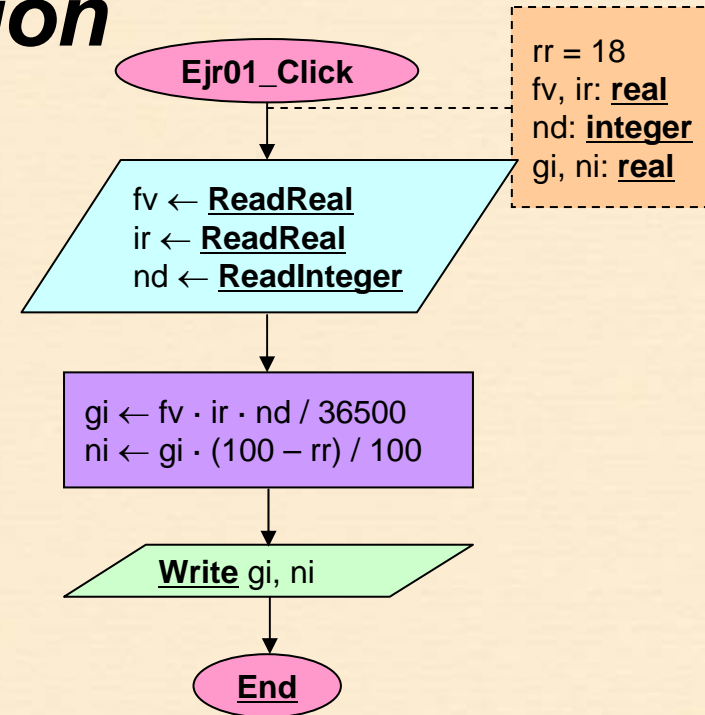
- Retention rate = 18% (**rr**)
- Days in a year = 365
- Face value (**fv**)
- Interest rate (**ir**)
- Number of days (**nd**)
- Gross interest (**gi**)
- Net interest (**ni**)

Ejr01: VB implementation

```

Sub Ejr01_Click ()
    Const rr As Double = 18
    Dim s As String
    Dim fv As Double, ir As Double
    Dim nd As Integer
    Dim gi As Double, ni As Double
    s = InputBox ("Face value: ")
    fv = Cdbl (s)
    s = InputBox ("Interest rate:")
    ir = Cdbl (s)
    s = InputBox ("Days:")
    nd = CInt (s)
    gi = fv * ir * nd / 36500
    ni = gi * (100 - rr) / 100
    MsgBox "Gross interest: " & CStr (ib) & vbCrLf & _
        "Net interest: " & CStr (it)
End Sub

```

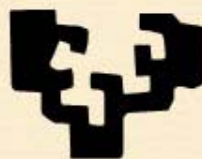




Escola Universitaria
de Ingenieria
Vitoria-Gasteiz

Ingeniaritzako
Unibertsitate Eskola
Vitoria-Gasteiz

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea