CSE 123
Introduction to Computing

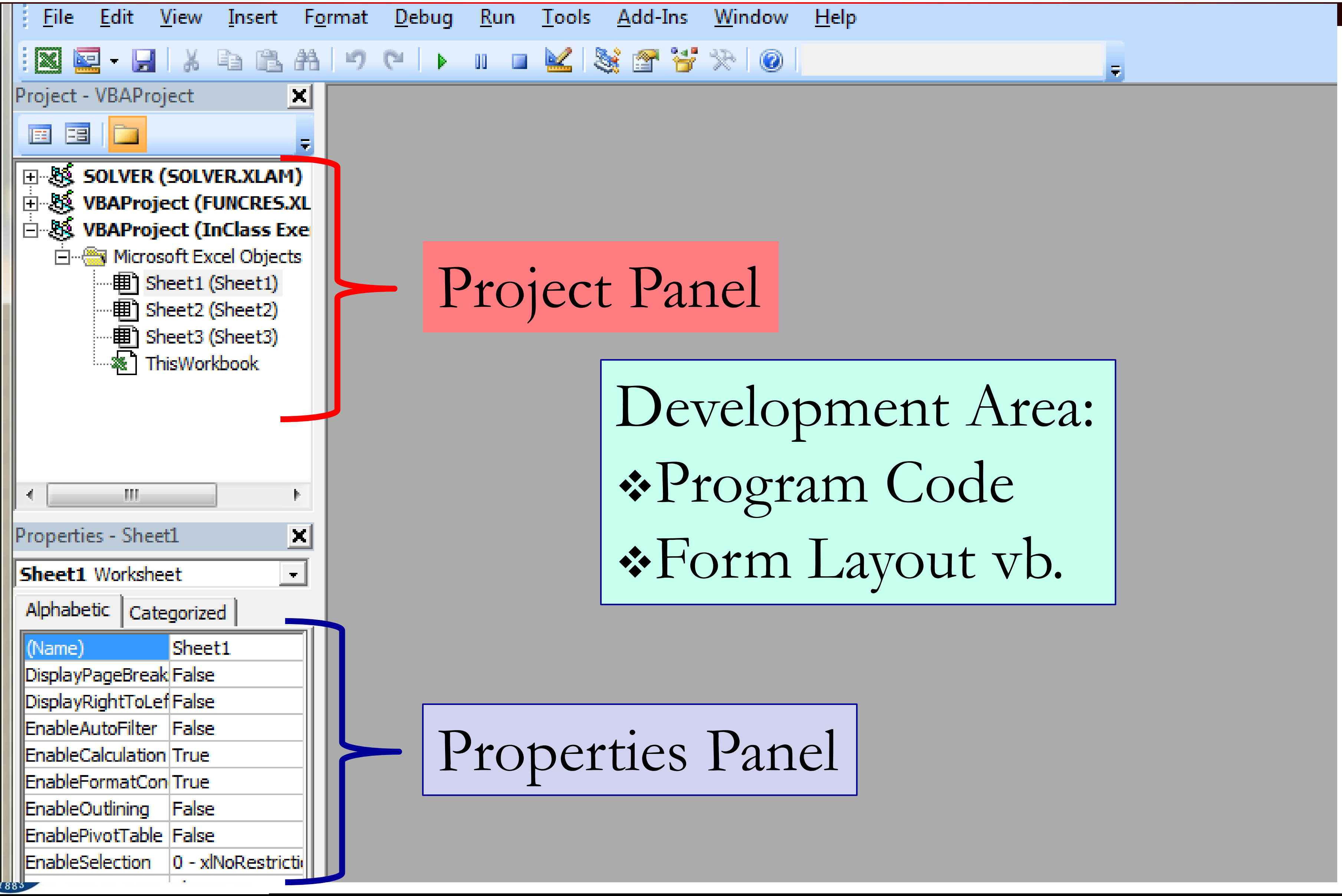
Lecture 6
Programming with VBA (Projects, forms, modules, variables, flowcharts)

SPRING 2012

Assist. Prof. A. Evren Tugtas

Marmara Üniversitesi

Starting with the VBA Editor



Project - VBAProject

- SOLVER (SOLVER.XLAM)
- VBAProject (FUNCRES.XL
- VBAProject (InClass Exe
  - Microsoft Excel Objects
    - Sheet1 (Sheet1)
    - Sheet2 (Sheet2)
    - Sheet3 (Sheet3)
    - ThisWorkbook

Properties - Sheet1

Sheet1 Worksheet

Alphabetic | Categorized

(Name)	Sheet1
DisplayPageBreak	False
DisplayRightToLef	False
EnableAutoFilter	False
EnableCalculation	True
EnableFormatCon	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestricti

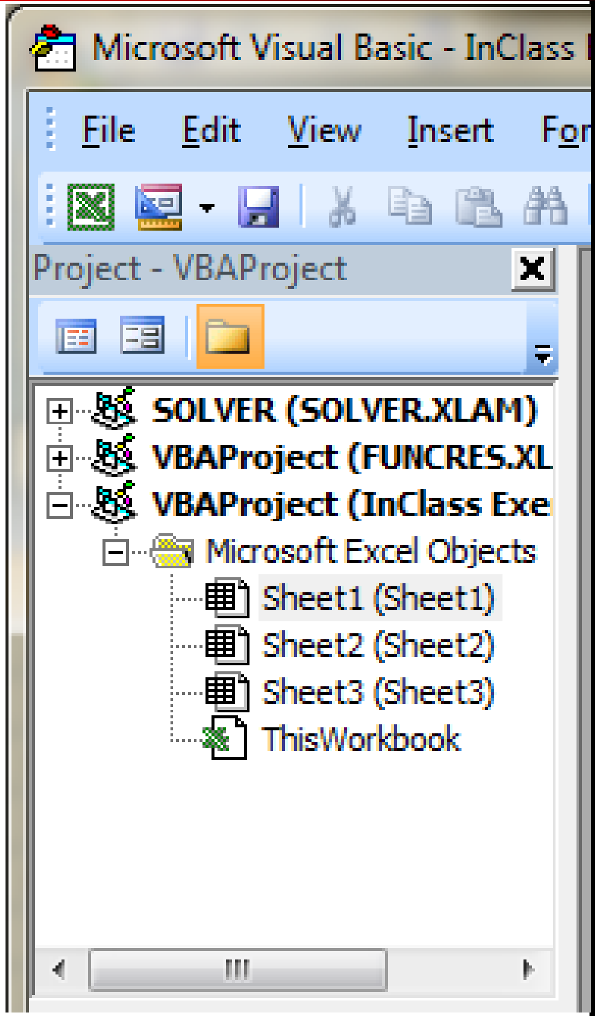
Project Panel

Development Area:
- ❖Program Code
- ❖Form Layout vb.

Properties Panel

Starting with the VBA Editor
Project panel

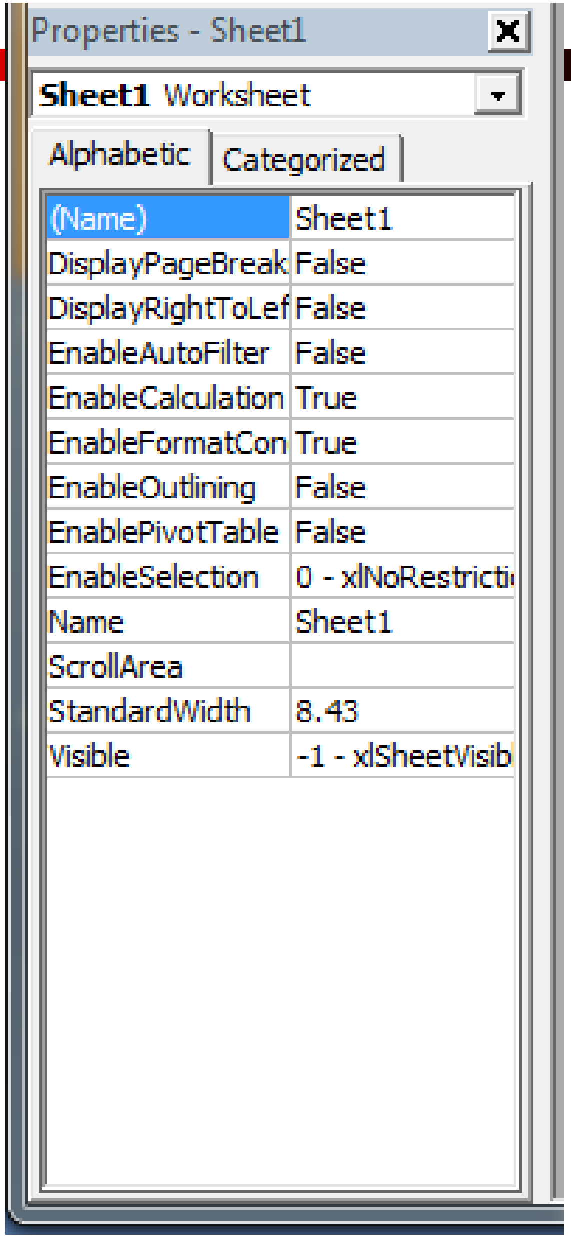
- List of items in a project
- Contains all of the
  - Functions
  - Forms
  - Subprograms etc.
- It also include default three sheets Sheet 1, 2, 3
- Each project item connects to a program code for that item only



SOLVER (SOLVER.XLAM)
VBAProject (FUNCRES.XL
VBAProject (InClass Exe
Microsoft Excel Objects
Sheet1 (Sheet1)
Sheet2 (Sheet2)
Sheet3 (Sheet3)
ThisWorkbook

Starting with the VBA Editor
Properties Panel

- Properties panel is used to
  - Access and modify various properties of the current selected object



(Name)	Sheet1
DisplayPageBreak	False
DisplayRightToLef	False
EnableAutoFilter	False
EnableCalculation	True
EnableFormatCon	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestricti
Name	Sheet1
ScrollArea	
StandardWidth	8.43
Visible	-1 - xlSheetVisibl

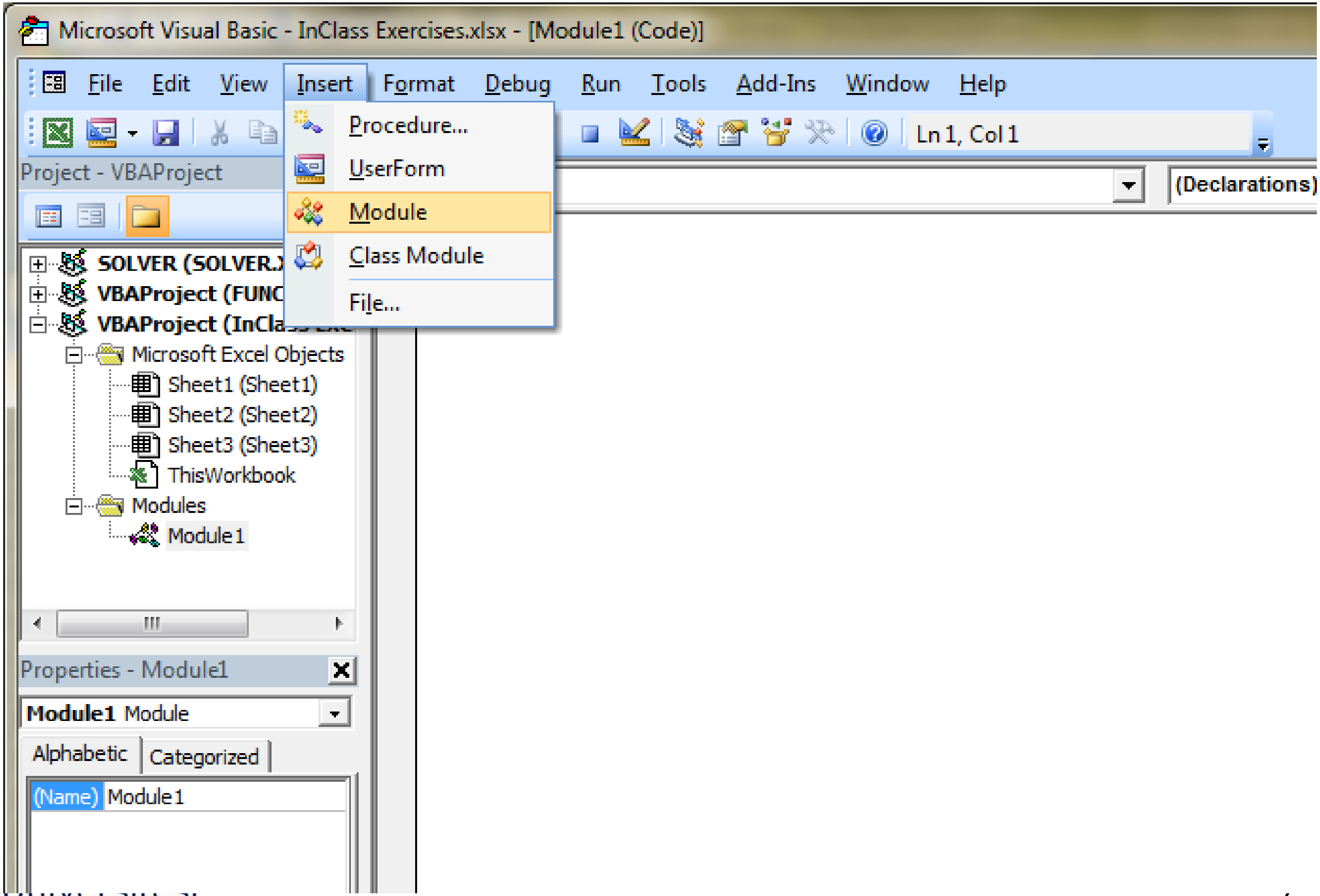
Starting with the VBA Editor
Development Area

- Development area is used to
  - Write a program code
  - View/Edit macros
  - Create/modify user forms

Projects, Forms, and Modules

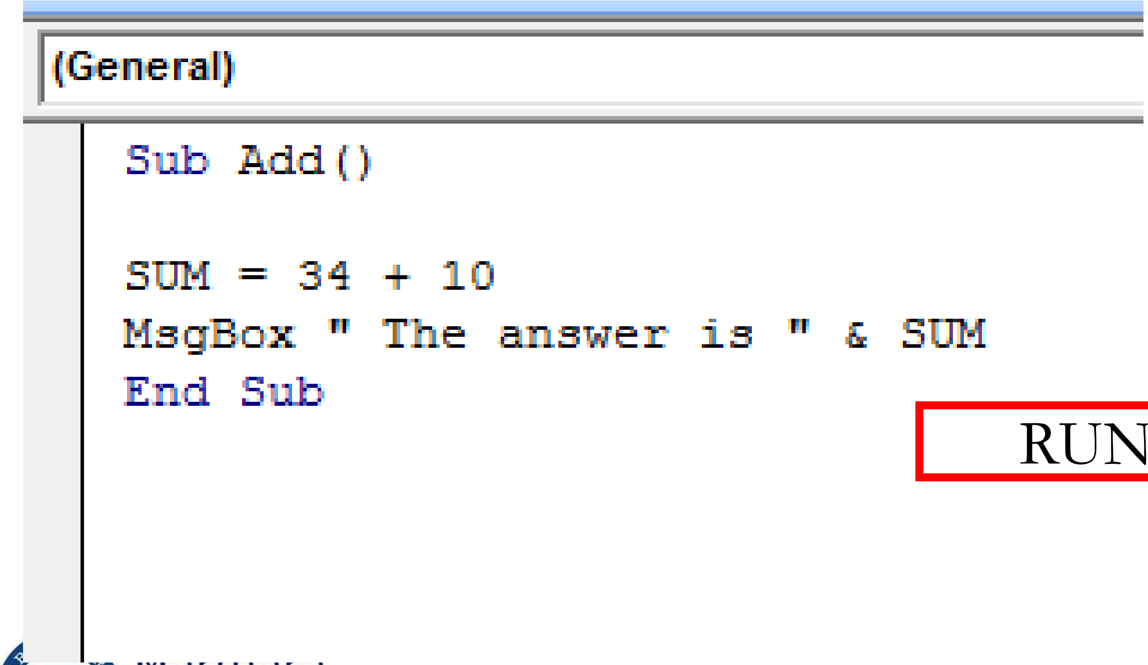
- A typical VB project includes
  - ***Forms*** to collect and present information to the user
  - ***Modules*** to hold variable definitions and program code
- A program code that is not specifically tied to an object (sheet, form, or form object) normally stored in a ***module***
- ***A module,*** stores program code such as variable definitions, subs and functions

Projects, Forms, and Modules



Sub procedures

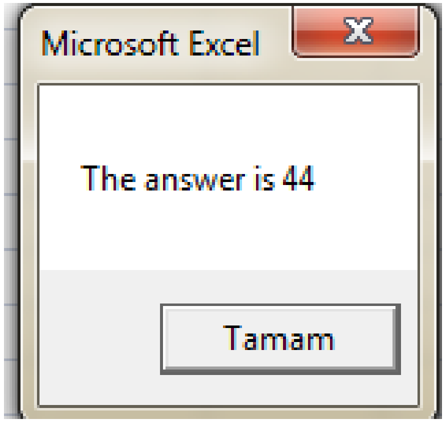
- A self contained unit of code
- All the macros are sub procedures
- Each Sub procedure begins with **Sub** ends with **End Sub** statement



```
Sub Add()

SUM = 34 + 10
MsgBox " The answer is " & SUM
End Sub
```

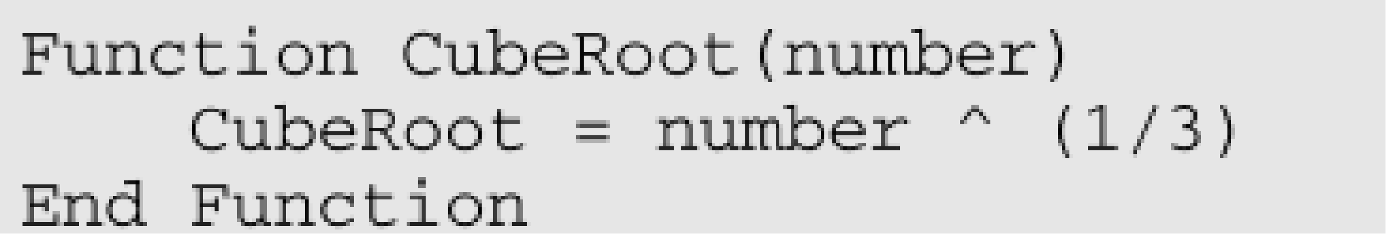
RUN



The answer is 44

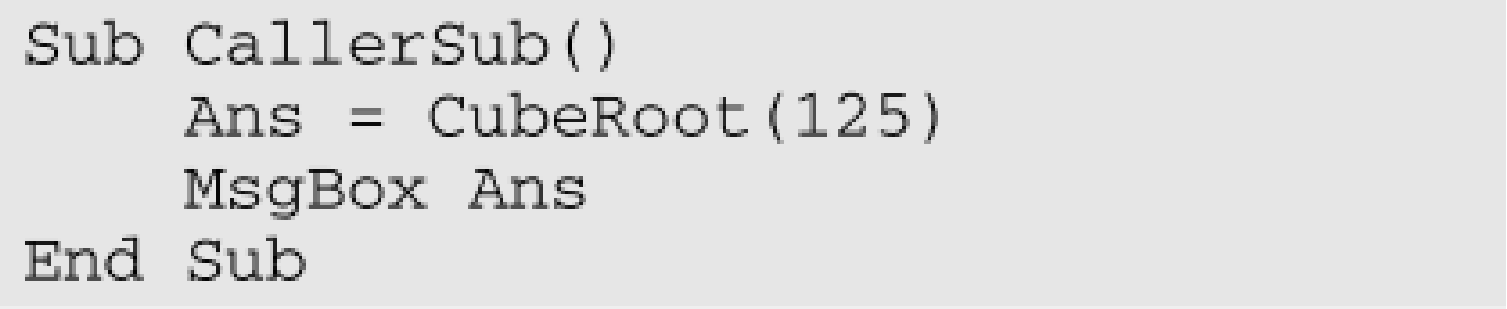
Function Procedure

- Performs calculations and returns a single value



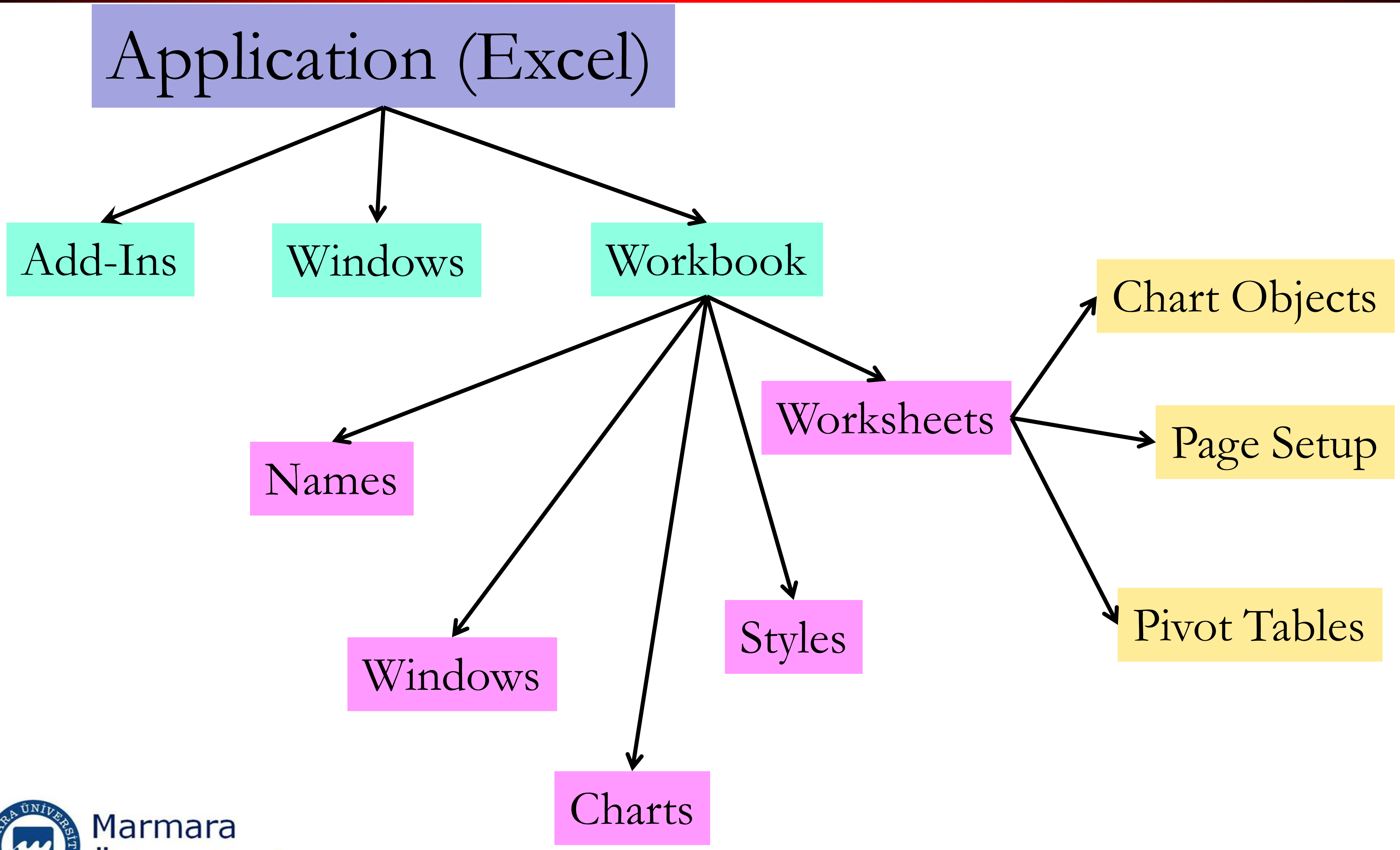
```
Function CubeRoot(number)
    CubeRoot = number ^ (1/3)
End Function
```

- You cannot execute a function directly, you need to call it from another procedure



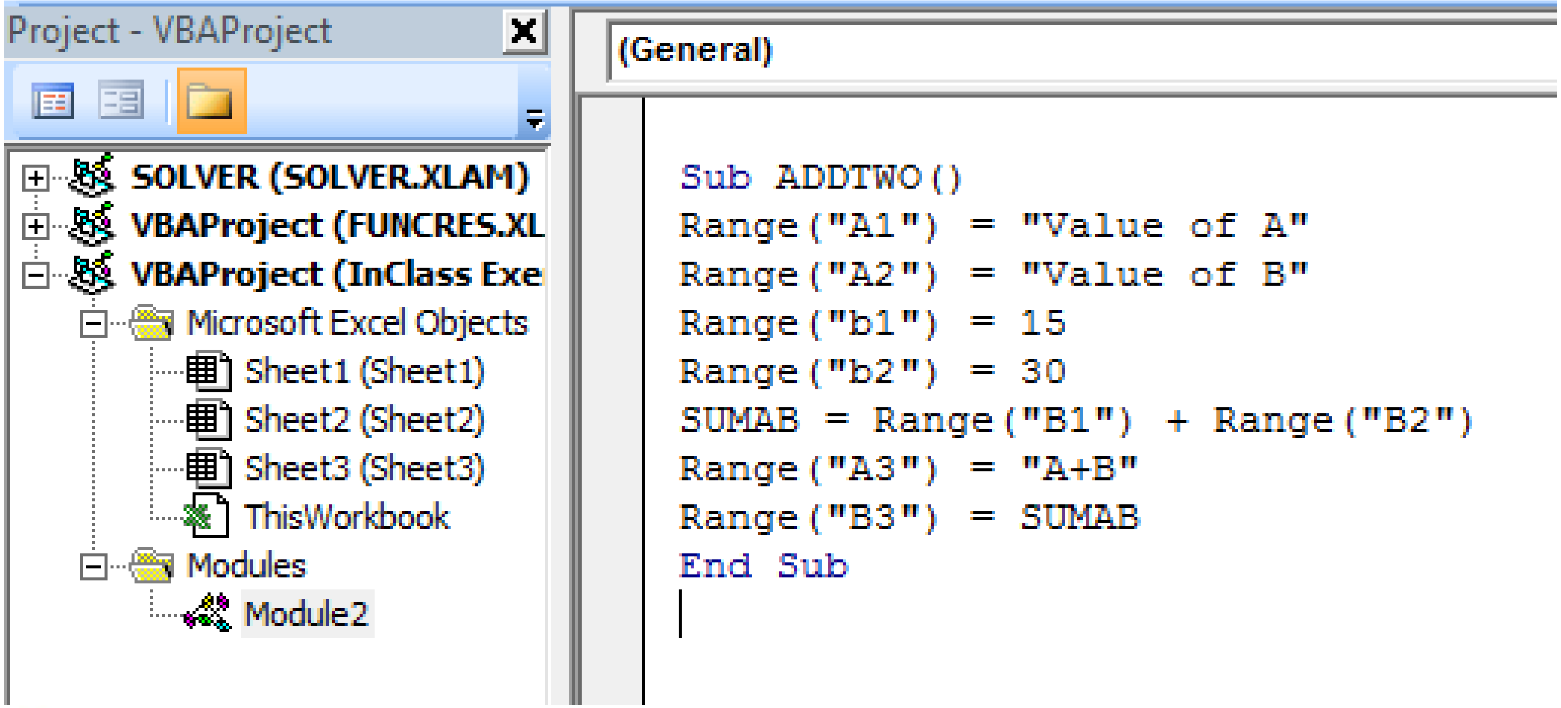
```
Sub CallerSub()
    Ans = CubeRoot(125)
    MsgBox Ans
End Sub
```

Objects


Application (Excel)
Add-Ins
Windows
Workbook
Chart Objects
Worksheets
Page Setup
Names
Pivot Tables
Styles
Windows
Charts

Object

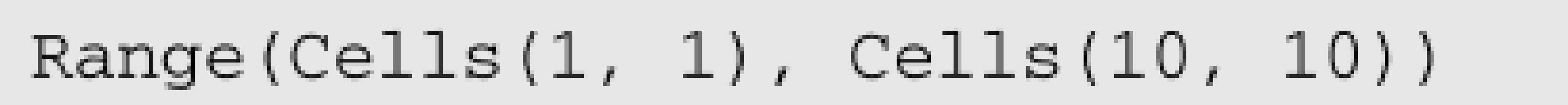
- Excel does not have a **Cell** object
- To manipulate a single cell
  - Use ***Range*** object with only one cell in it



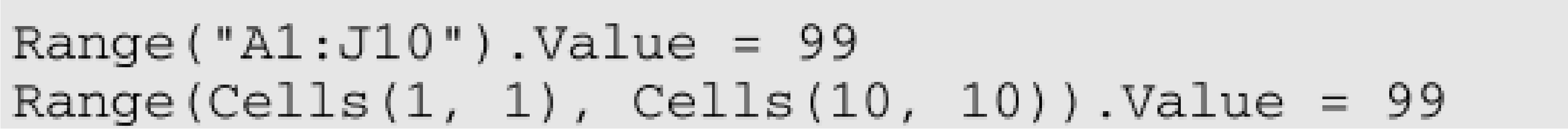
```
Sub ADDTWO()
Range("A1") = "Value of A"
Range("A2") = "Value of B"
Range("b1") = 15
Range("b2") = 30
SUMAB = Range("B1") + Range("B2")
Range("A3") = "A+B"
Range("B3") = SUMAB
End Sub
```

Marmara Üniversitesi

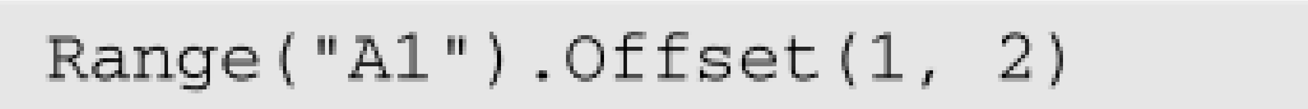
Range



```
Range(Cells(1, 1), Cells(10, 10))
```



```
Range("A1:J10").Value = 99
Range(Cells(1, 1), Cells(10, 10)).Value = 99
```

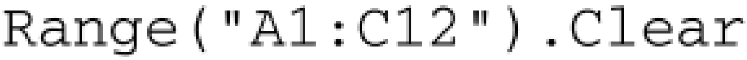


```
Range("A1").Offset(1, 2)
```

below right

Methods

- Objects also have methods
- Methods are actions taken within an object
- ***Range*** objects have a ***Clear*** method
- Methods are connected to an object with a dot.



```
Range("A1:C12").Clear
```

- Activate
- Close
- Save
- Save as

Methods

Variables

- Excel does not require you to declare varibles using the Dim statement before you use them in your code.
- However, it is useful to declare variables to prevent errors due to misspelling
- Suppose you declare variable called BOD in your code and later you misspelled the name as BOS, if you did not use a Dim statement, the compiler will not flag it as an error and will create a new variable called BOS, which will cause an error in your code.

Variables

- Option Explicit
- This statement requires that all variables be declared using a Dim statement
- If you have ***Options Explicit*** statement at the top, you have to declare all your variables.
- If you declare your variables using a Dim statement, misspelled names will be flagged as error by the compiler.

e.g.

Dim J, K, L As Single

Variables

Single

- Single presicion real numbers (4 bytes)
- -3.402823E38 to 3.402823E38
- Used for general low presicion math

- Examples 10.23, 3412.90, -230.23

Variables

Double

- Double presicion real numbers (8 bytes)
- -1.7976931348623**E**308 to 1.7976931348623**E**308
- Used for general high presicion math

- Example 10.2334234

Variables

Integer

- Small integer numbers (2 bytes)
- -32768 to 32768
- Used for counters, index values
- Example 1   3   8900   76393

Variables

Long

- Long integer numbers (4 bytes)
- -2147483648 to 2147483648
- Used for counters, index values for long numbers

- Example 1280000

Variables

Boolean

- Logical values (2 bytes)
- TRUE or FALSE
- Status variables

Date

- Date values (8bytes)
- Used for dates and times

Variables

String

- Text strings
- "the result is:"
- "C:\my Documents"

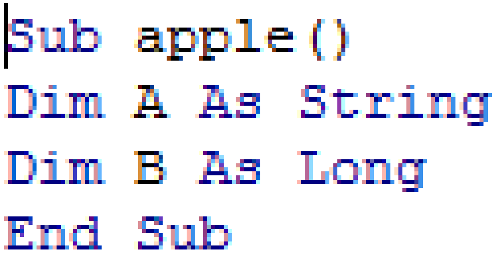
- Used for words, phrases, file names

Variables

- Sometimes you need to access a variable from an outside procedure in which the variable is declared. In this case you need to declare a wider scope variable
    - Public
    - Private
    - Procedure

Variables

- Procedure :
- Available only to procedure that contains it. Used oly for the variables that operate in the procedure in which they are declared.
- Implicit declarations are automatically assigned as a procedure scope

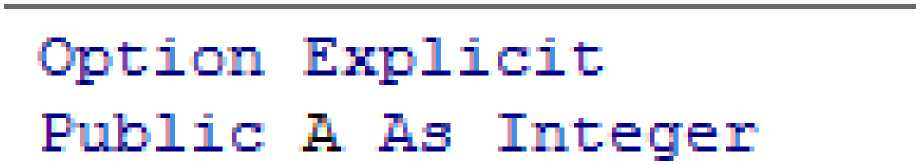


```
Sub apple()
Dim A As String
Dim B As Long
End Sub
```

You cannot pass A and B to other procedures

Variables

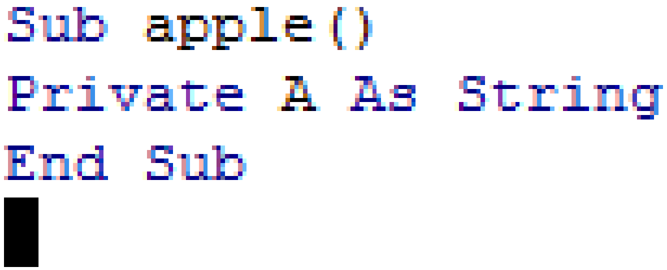
- Public :
- Available to all procedures in all modules in the project that contains it



```
Option Explicit
Public A As Integer
```

Variables

- Private :
- Available to all procedures in the module that contains it, but not to procedures in other modules.

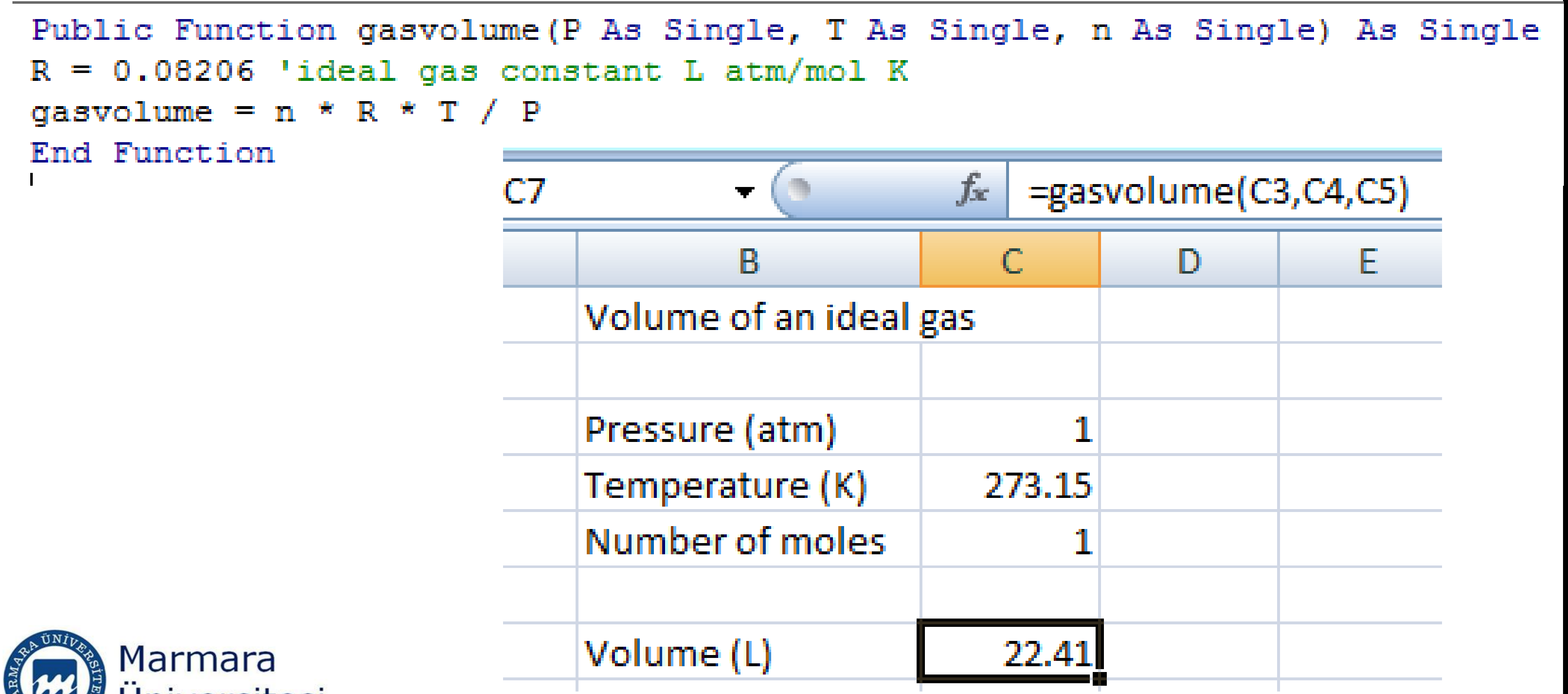


```
Sub apple()
Private A As String
End Sub
```

VBA – Example

- Calculate the volume occupied by an ideal gas

$$PV=nRT$$



```
Public Function gasvolume(P As Single, T As Single, n As Single) As Single
R = 0.08206 'ideal gas constant L atm/mol K
gasvolume = n * R * T / P
End Function
```

C7 =gasvolume(C3,C4,C5)

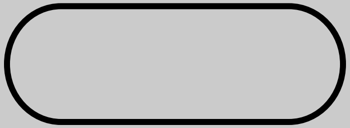
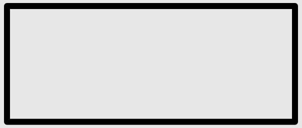
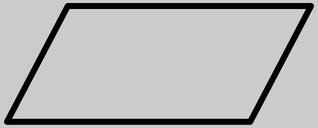
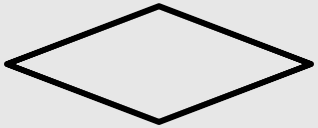
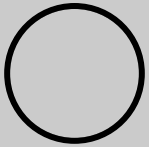
B	C	D	E
Volume of an ideal gas			
Pressure (atm)	1		
Temperature (K)	273.15		
Number of moles	1		
Volume (L)	22.41		

Marmara Üniversitesi

Flowcharts

- A flowchart is a visual depiction of a program's operation
- It is designed to show what a program does in a step by step fashion
- It is the programmers assistin the development of a program
- There are standard symbols used in computer flowcharts.

Flowcharts

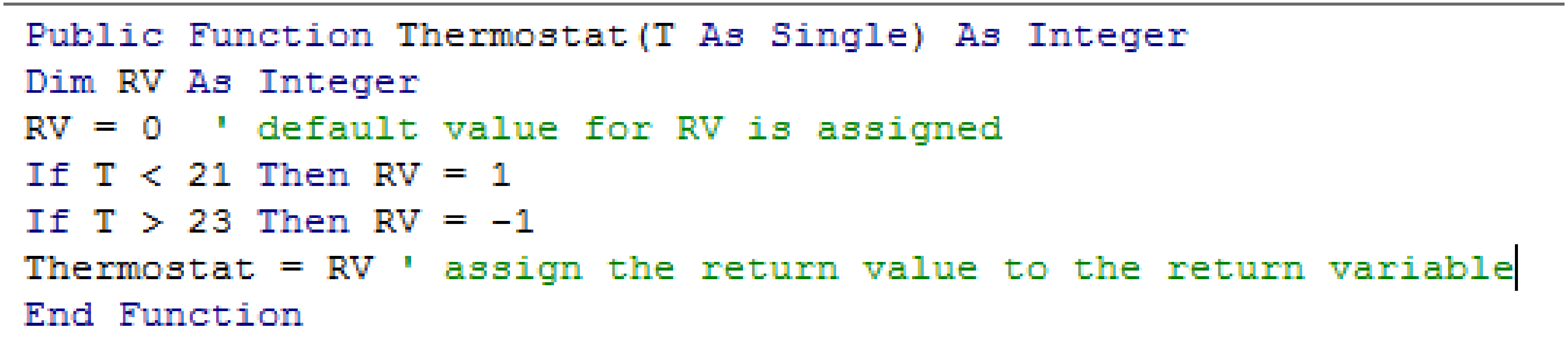
SYMBOL	NAME	USAGE
	Terminator	Indicates the start or end of a program
	Operation	Indicates a computation step
	Data	Indicates and input or output step
	Decision	Indicates a decision point in a program
	Connector	Indicates that the flowchart continues in another location

Example

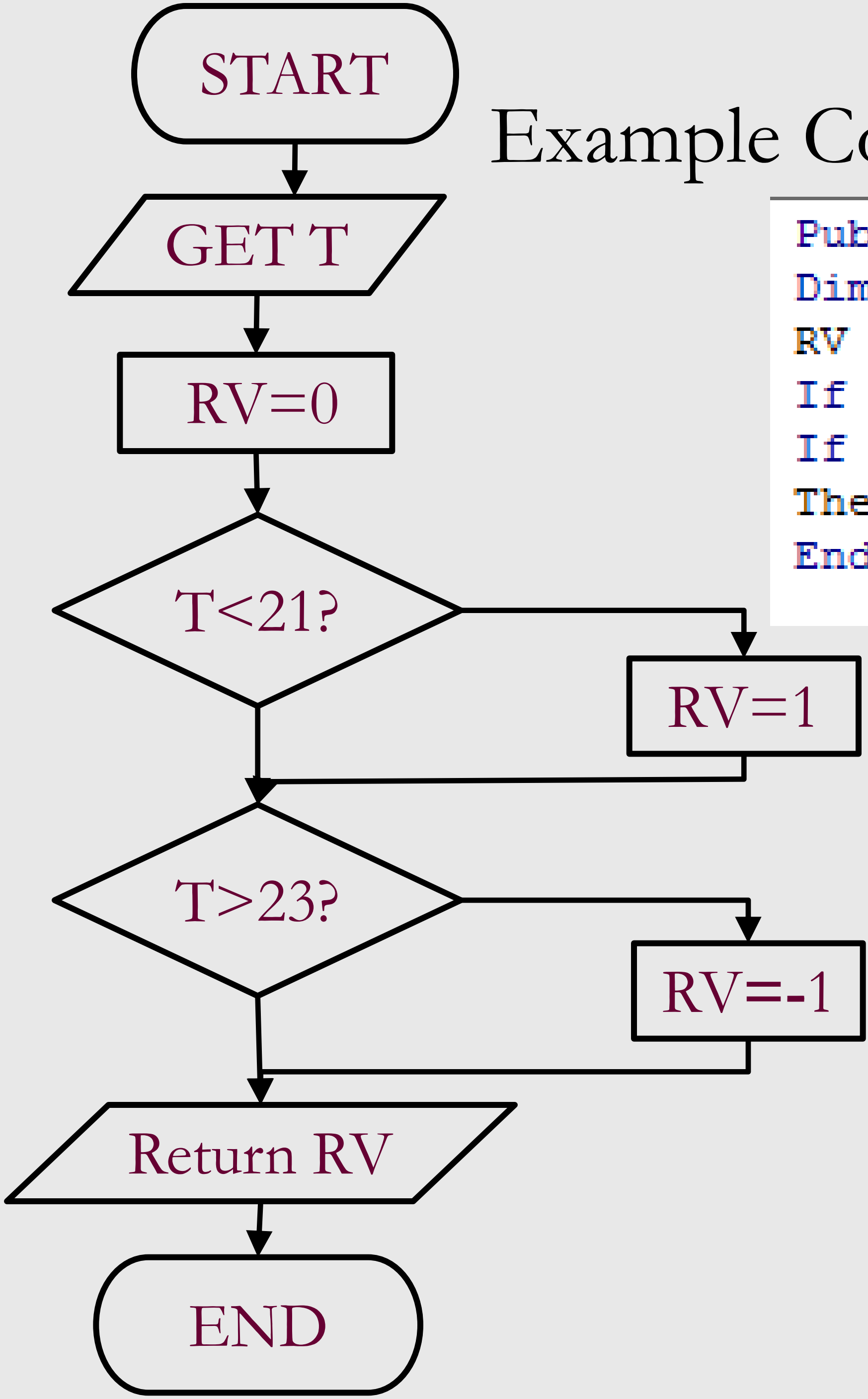
Consider a thermostat function that is designed to return a code value to turn a heater on or off

- If temperature is below 21^{o}C, return the value 1 to indicate the heater should be activated
- If temperature is above 23^{o}C, return the value -1 to indicate that the heater should shut off
- If the temperature is between 21^{o}C and 23^{o}C, return the value 0 to indicate that there should be no change in heaters status

Example Cont’d, Code

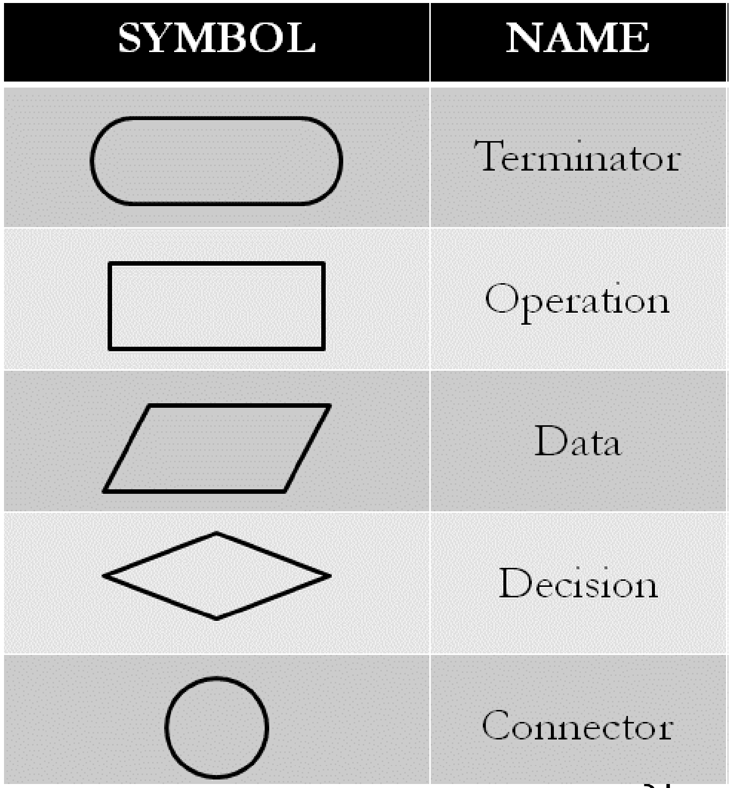


```
Public Function Thermostat(T As Single) As Integer
Dim RV As Integer
RV = 0  ' default value for RV is assigned
If T < 21 Then RV = 1
If T > 23 Then RV = -1
Thermostat = RV ' assign the return value to the return variable
End Function
```



Example Cont'd, Flowchart

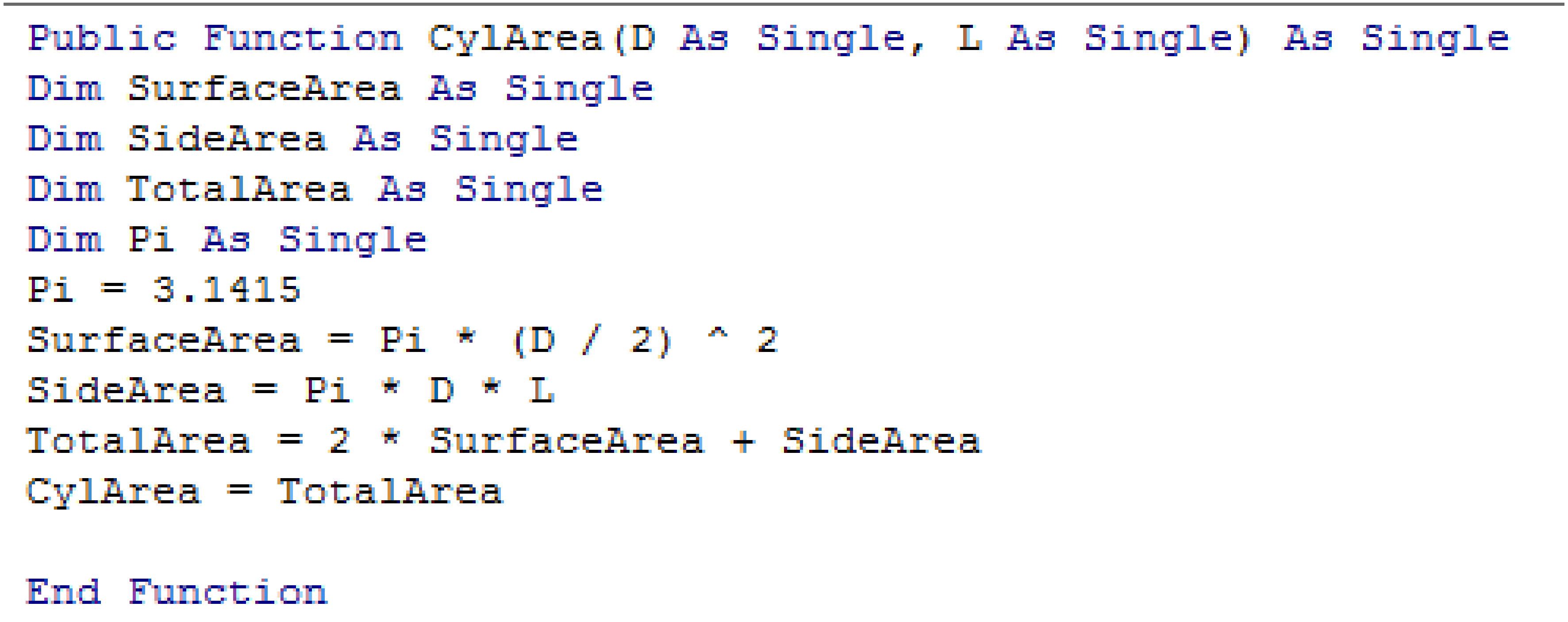
```
Public Function Thermostat(T As Single)
Dim RV As Integer
RV = 0   ' default value for RV is assig
If T < 21 Then RV = 1
If T > 23 Then RV = -1
Thermostat = RV ' assign the return val
End Function
```



SYMBOL	NAME
▭	Terminator
□	Operation
▱	Data
◇	Decision
○	Connector

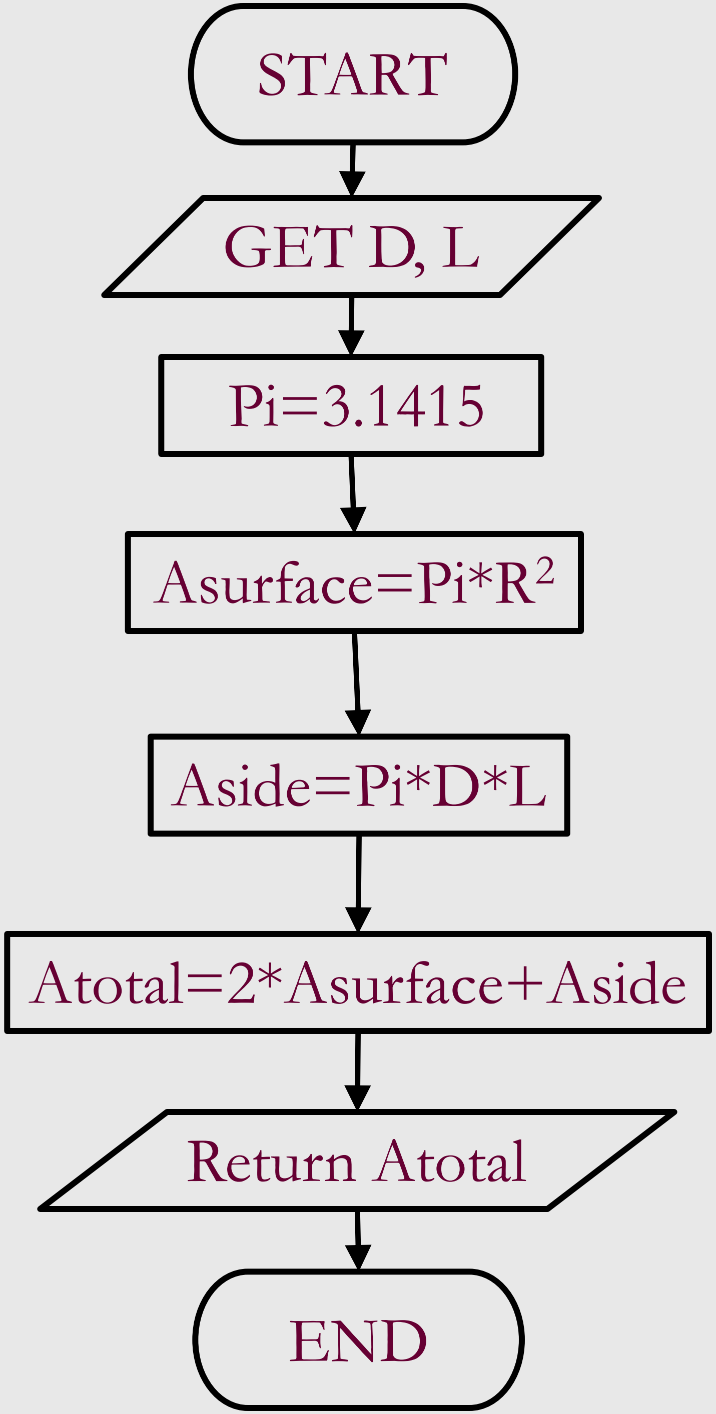
Example

Calculate the area of a cylinder



```
Public Function CylArea(D As Single, L As Single) As Single
Dim SurfaceArea As Single
Dim SideArea As Single
Dim TotalArea As Single
Dim Pi As Single
Pi = 3.1415
SurfaceArea = Pi * (D / 2) ^ 2
SideArea = Pi * D * L
TotalArea = 2 * SurfaceArea + SideArea
CylArea = TotalArea

End Function
```



START

GET D, L

Pi=3.1415

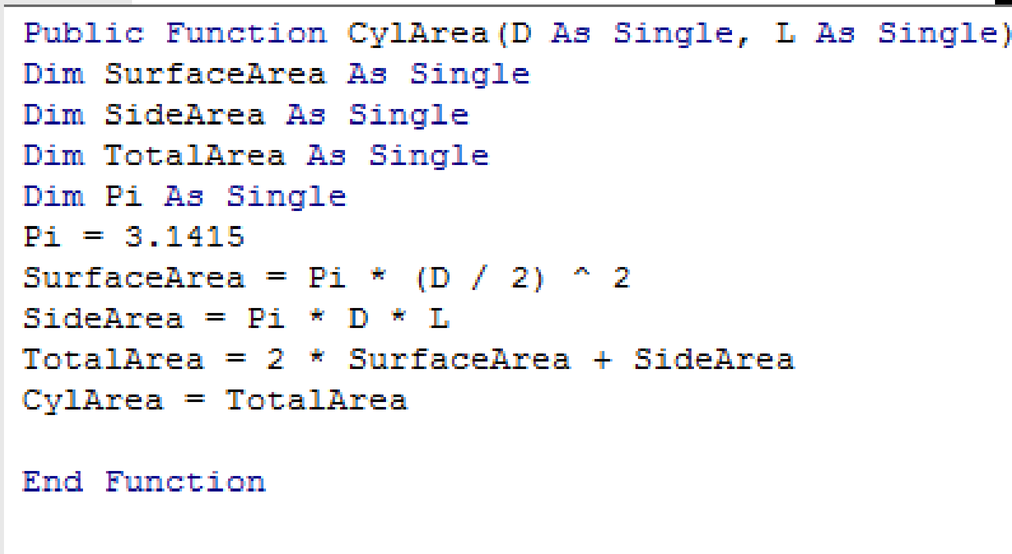
Asurface=Pi*R^2

Aside=Pi*D*L

Atotal=2*Asurface+Aside

Return Atotal

END



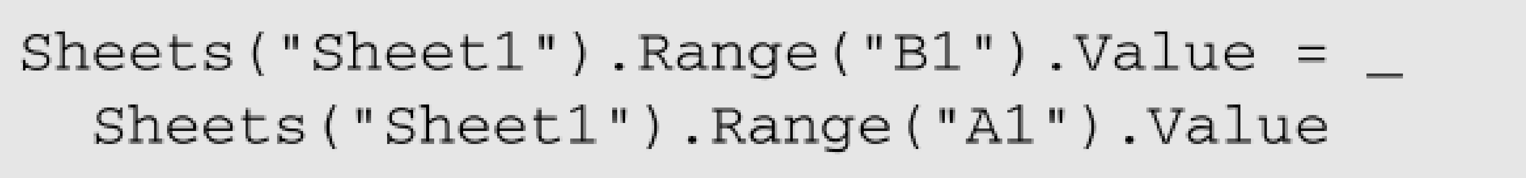
```
Public Function CylArea(D As Single, L As Single)
Dim SurfaceArea As Single
Dim SideArea As Single
Dim TotalArea As Single
Dim Pi As Single
Pi = 3.1415
SurfaceArea = Pi * (D / 2) ^ 2
SideArea = Pi * D * L
TotalArea = 2 * SurfaceArea + SideArea
CylArea = TotalArea

End Function
```

SYMBOL	NAME
⬭	Terminator
▭	Operation
▱	Data
◇	Decision
○	Connector

Programming Tips

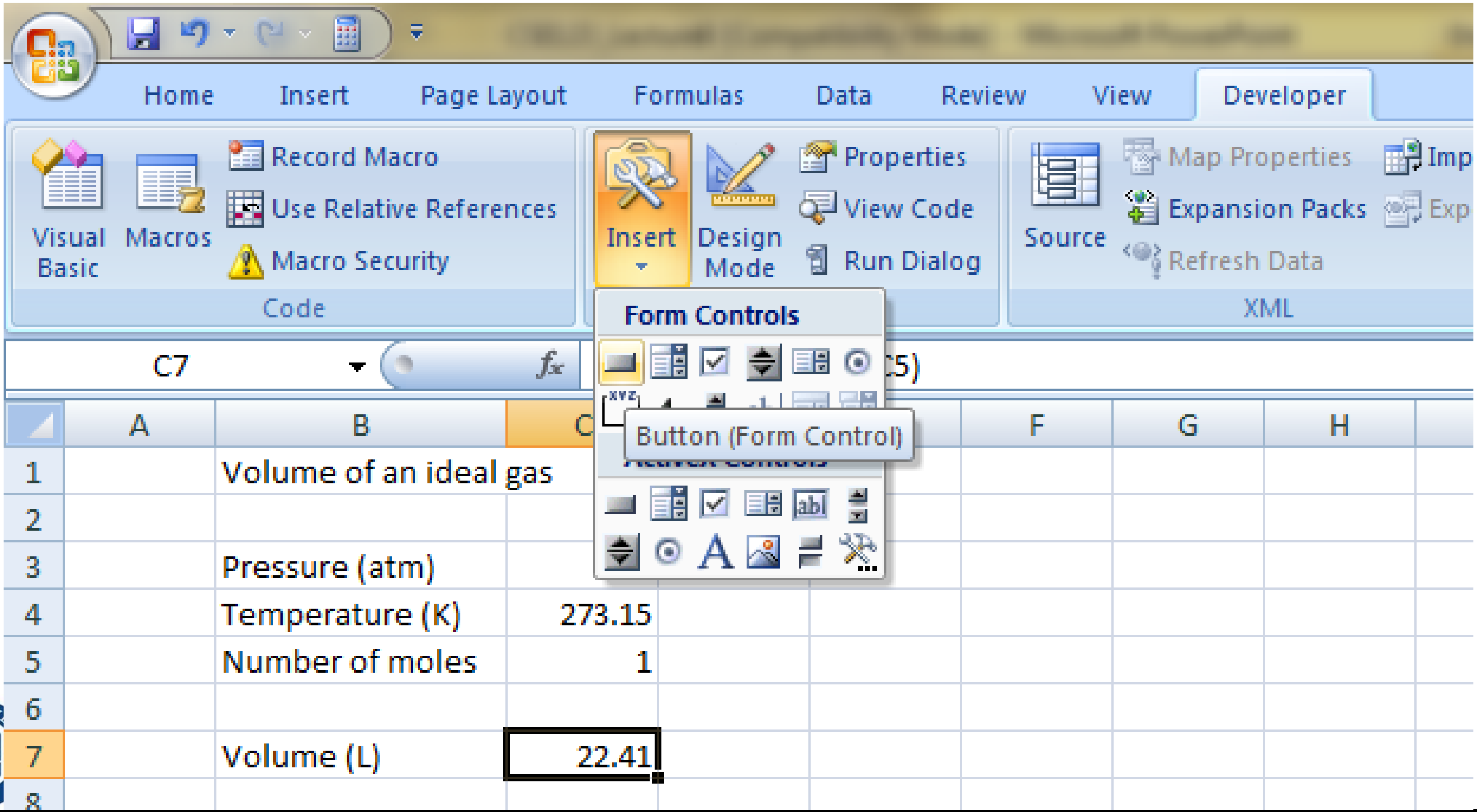
- If the code line is too long you can break the statement into two lines



```
Sheets("Sheet1").Range("B1").Value = _
  Sheets("Sheet1").Range("A1").Value
```

Executing a Sub Procedure from a button

- Add button from
- Developer/Insert/Form Controls
- Assigne button to the to a macro



	A	B	C
1		Volume of an ideal gas	
2			
3		Pressure (atm)	
4		Temperature (K)	273.15
5		Number of moles	1
6			
7		Volume (L)	22.41