



Bringing British  
Education to You  
[www.nccedu.com](http://www.nccedu.com)

# Office Solutions Development

*Topic 5:*

*An Introduction to VBA and Macros*

# Scope and Coverage

*This topic will cover:*

- Macros - what they can be used for, how they can be developed and the role that VBA plays in their development

# Learning Outcomes - 1

*By the end of this topic, students will be able to:*

- Define what is meant by macro and VBA
- Explain that there are range of macros that are used for different purposes
- Describe the methods that can be used to develop macros
- Explain the issues of macros and security

# Learning Outcomes - 2

*By the end of this topic, students will be able to:*

- Use the Visual Basic environment to create macros
- Use the Record Macro facility
- Save macros
- Edit macros

# Terminology

- Terminology will be explained in the lecture, tutorial and laboratory session. You should take notes!
- Ask questions if you do not understand

# What is a Macro?

- A **macro** is made up of a series of instructions which are recorded so that they can be played back later and this can be done by a single keystroke or command. These commands can act as a shortcut to enable repetitive tasks to be completed more efficiently and quickly.
- They can be run from a button, toolbar or menu.
- **Event macros** are started by an event, such as double clicking a cell, opening a workbook, or activating a worksheet.

# Why are Macros Used?

- Macros can be used to automate and add functionality to many tasks, such as:
  - formatting whole or parts of documents, worksheets,
  - graphs
  - re-organising data
  - templates
  - formatting and reports
  - changing values of stored data
  - error message notification
  - interactive user forms/boxes, e.g. data entry

# How are Macros Developed?

- There are two ways to develop macros:
  - The easiest method is to use the Macro Recorder to record a sequence of instructions selected by a user. An application, such as Excel or Word Excel, stores information about each of the steps taken by a user as they record their commands. The macro can then be run to repeat these commands.
  - Another method is for a user to write instructions using the VBA programming language.



# What is VBA?

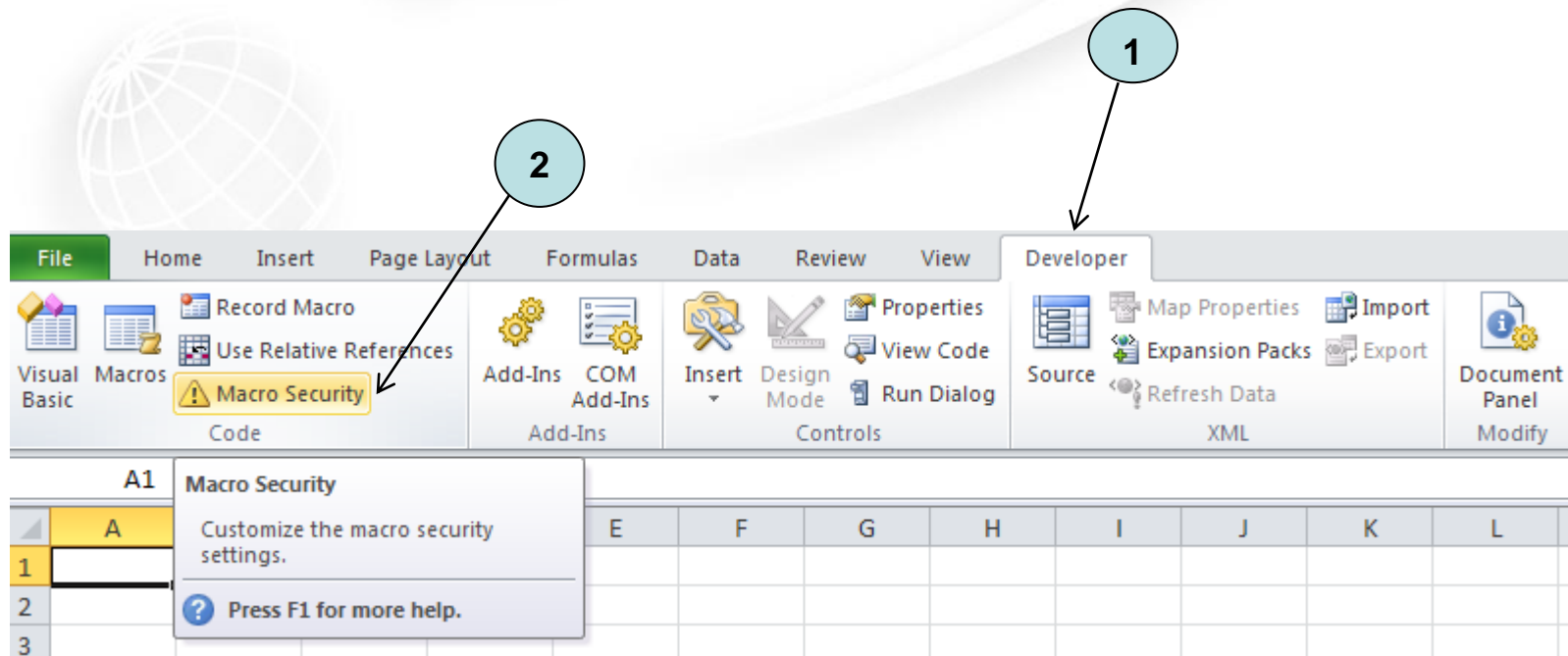
- VBA stands for **Visual Basic for Applications** and is a programming language.
- It has its own **integrated development environment** within Microsoft Office called the **VB (Visual Basic) Editor**.
- Macros can be developed using it, their instructions written in the VB Editor instead of being recorded by the Macro Recorder.
- Note that when a macro is developed using a Macro Recorder, a VBA program is written in the background and this program can be read in the VB Editor.

# Macros and Security

- Macros are subject to security problems, such as viruses, which are frequently attached to software downloads hidden in the code of macros.
- Always ensure that your security settings are set appropriately – see <http://office.microsoft.com/en-us/>

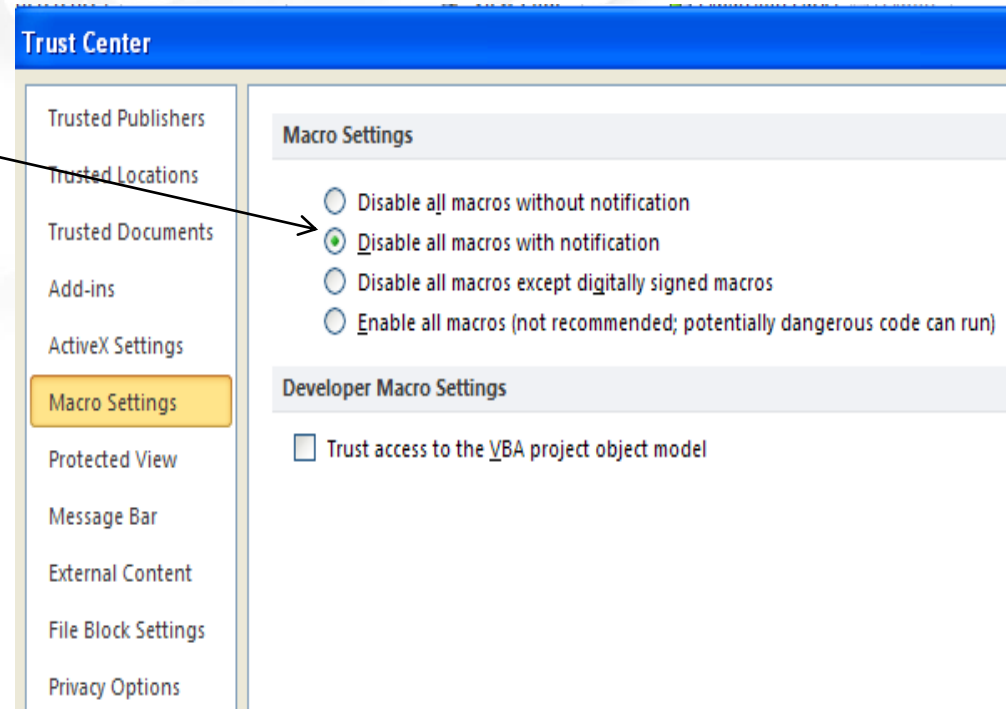
# Applying Security Levels

- Apply security settings by selecting the following:



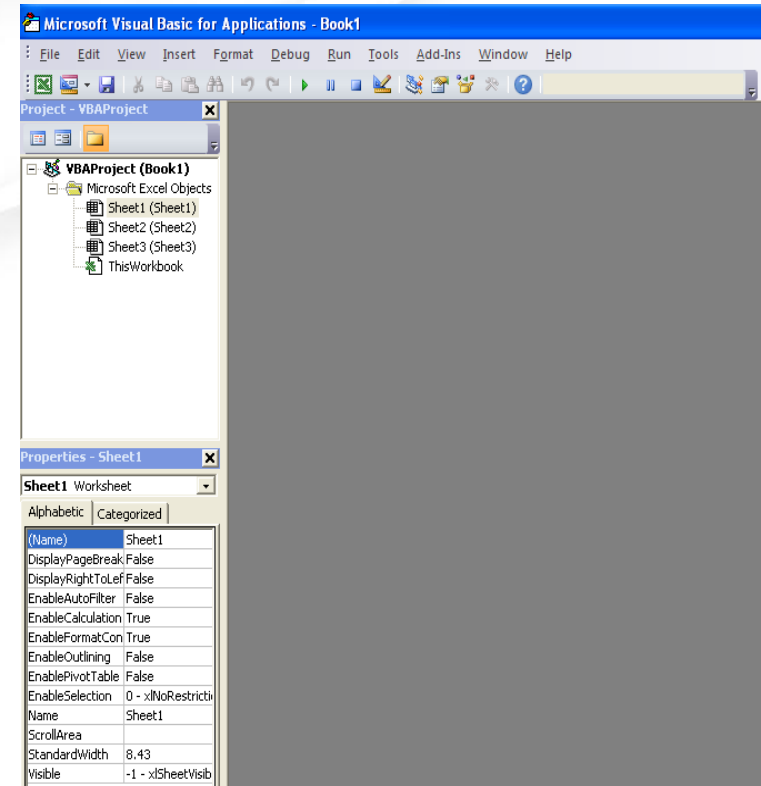
# Selecting Security Levels

- The second selection is generally recommended.



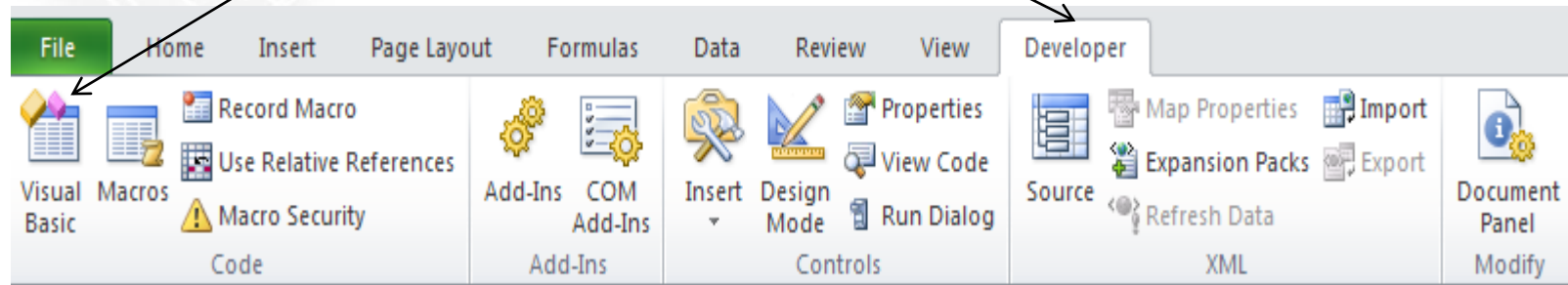
# The Visual Basic Environment

- The Visual Basic Editor is a program within Excel and Word that allows you to communicate with Excel and Word by writing specific instructions or code.




# Selecting the VB Editor

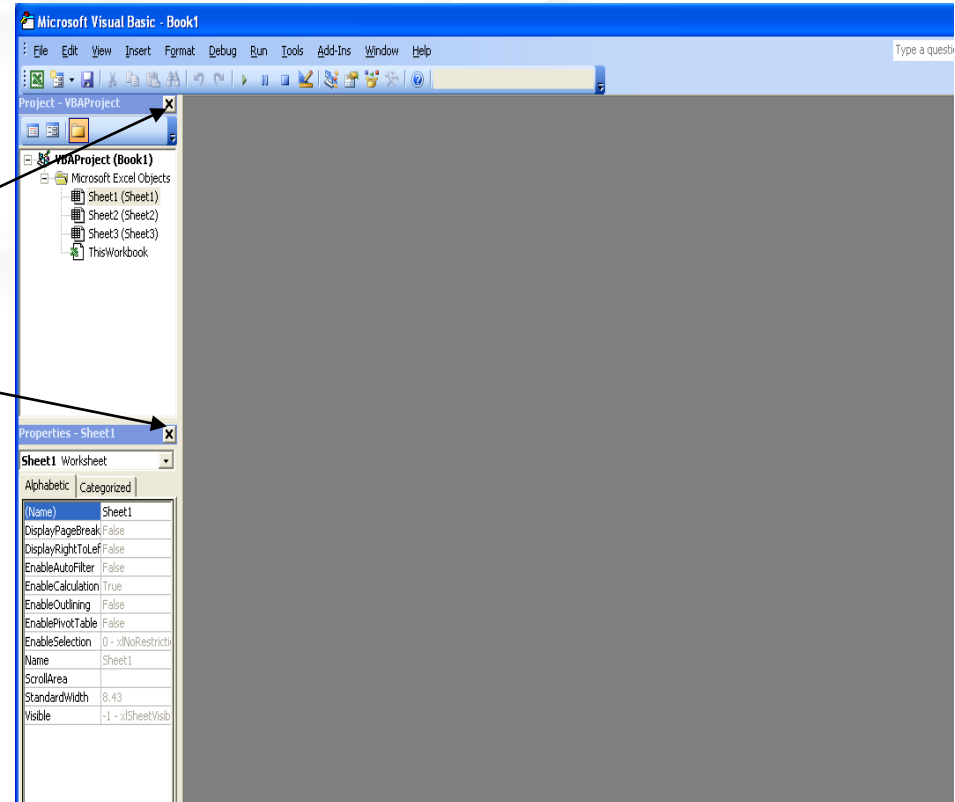
- This can be done either by:
  - Selecting the *Developer* option from the Ribbon
  - Selecting Visual Basic



- Or simply pressing the **ALT key** and the **F11 key**

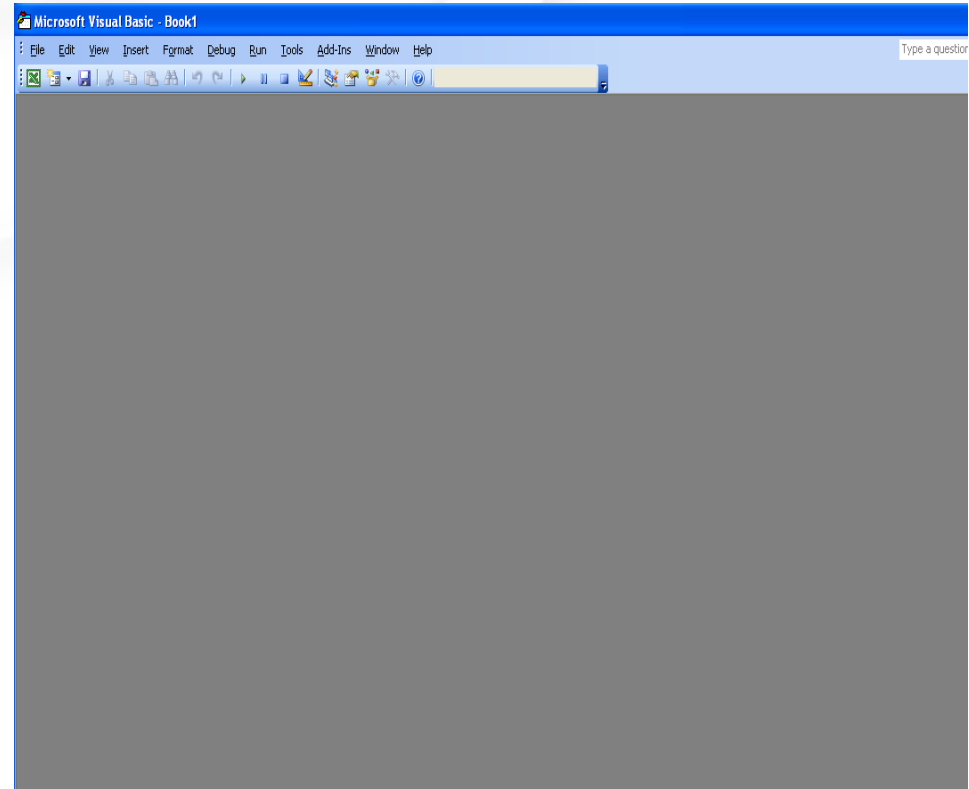
# The VB Editor

- This is the VB Editor window.
- Click on the two  to close the two windows if they are open.



# The VB Editor Screen

- The VB editor screen will now be displayed.
- To work efficiently in the VB Editor, there are three windows that should be displayed and they can be set up as follows:



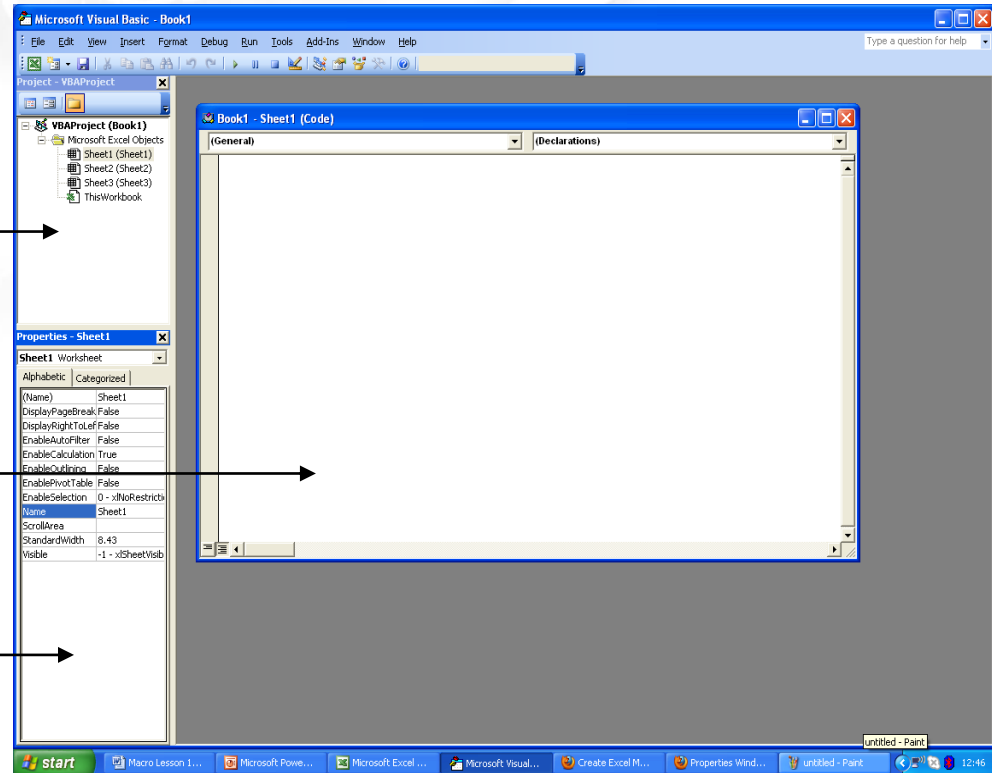


# The VB Editor Windows

- *Project Window  
(Project Explorer)*

- *Code Window*

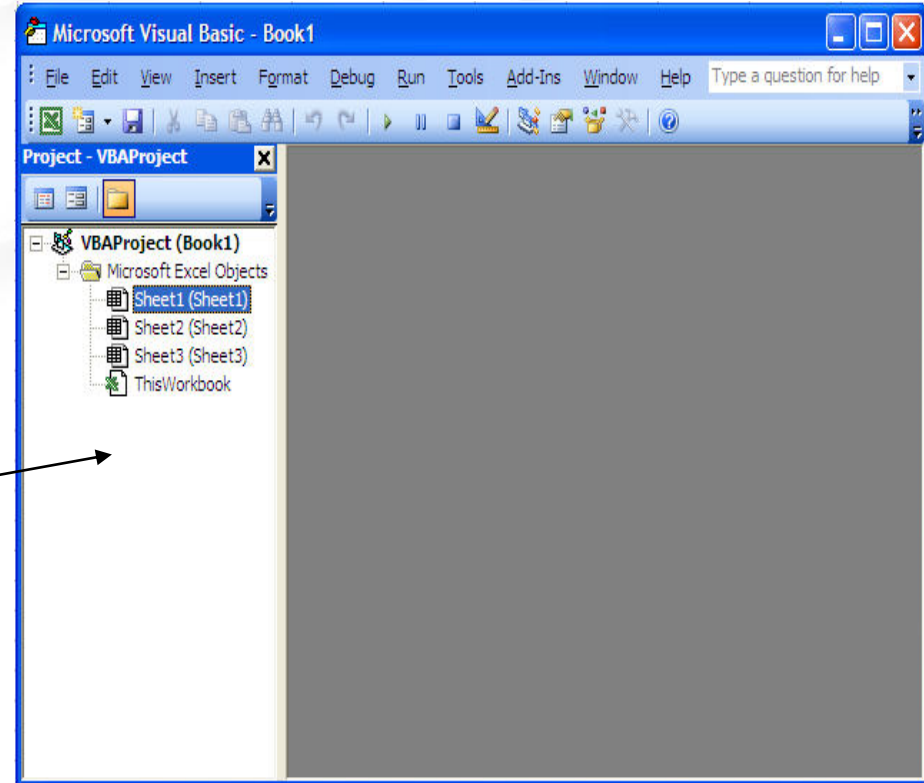
- *Properties Window*



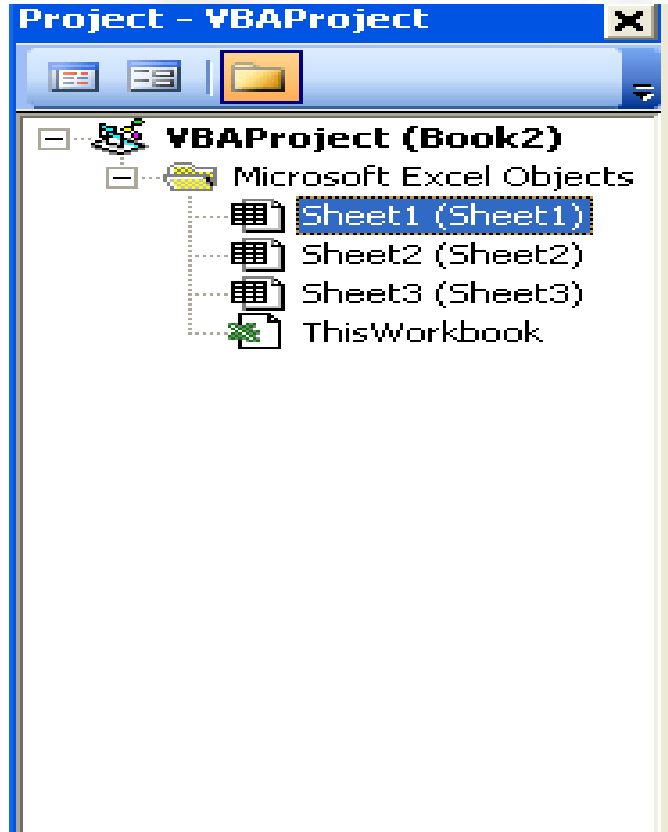
We will discuss each in turn

# Setting up the Project Window

- Select *View* from the main menu bar.
- Select *Project Explorer*
- The Project Window will be displayed.



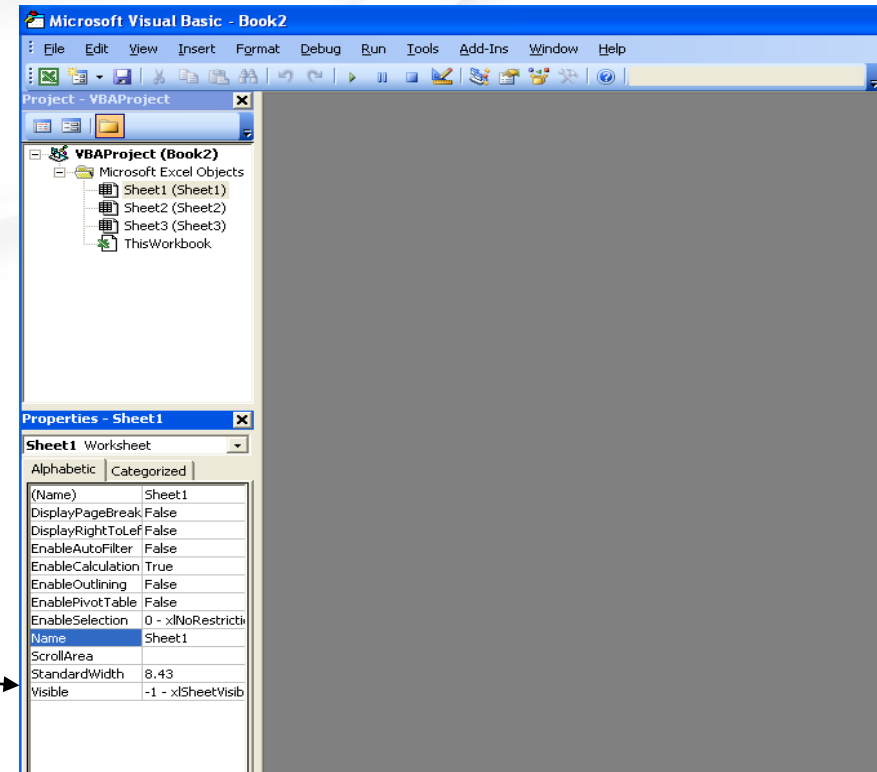
# The Project Window



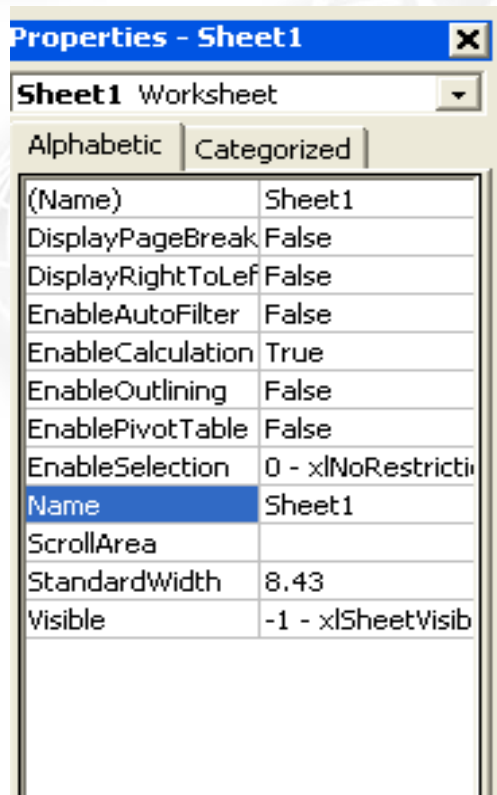
- This lists the projects and sheets (modules) and user forms.

# Setting up the Properties Window

- Select *View* from the main menu bar
- Select *Properties Window*
- The Properties Window will be displayed



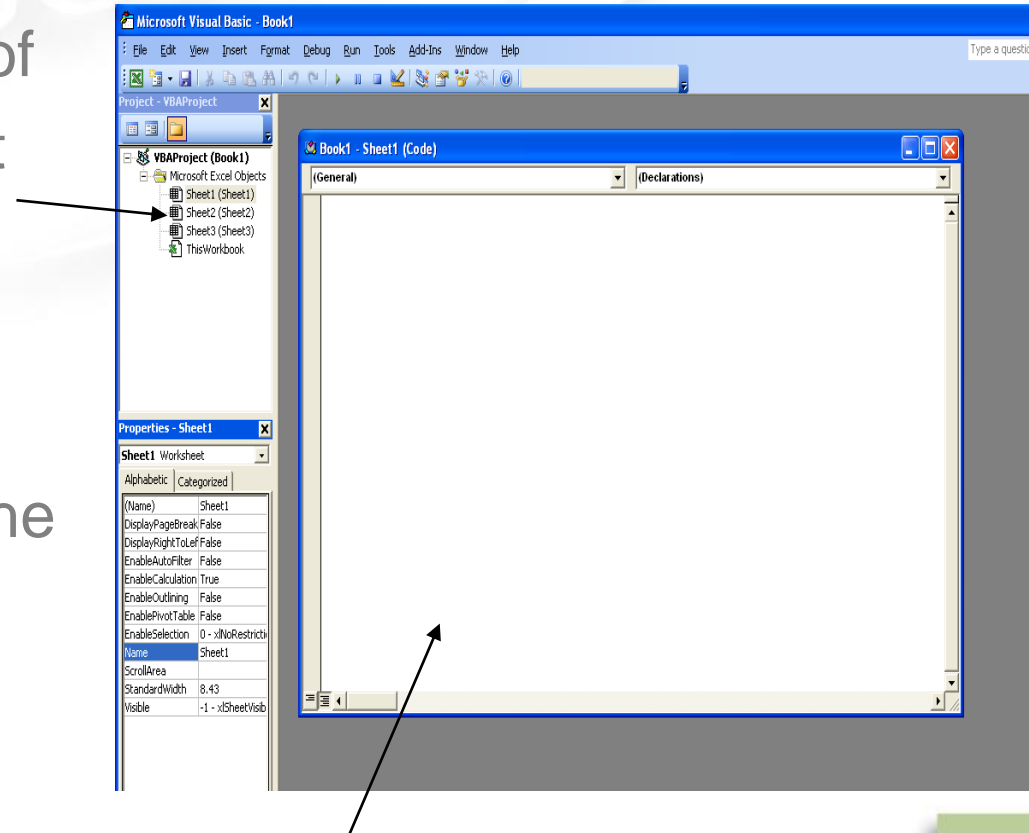
# The Properties Window



- This displays the properties of the **object** that is selected, e.g. the worksheet or the properties of control commands, e.g. command buttons, list boxes, etc. on user forms.

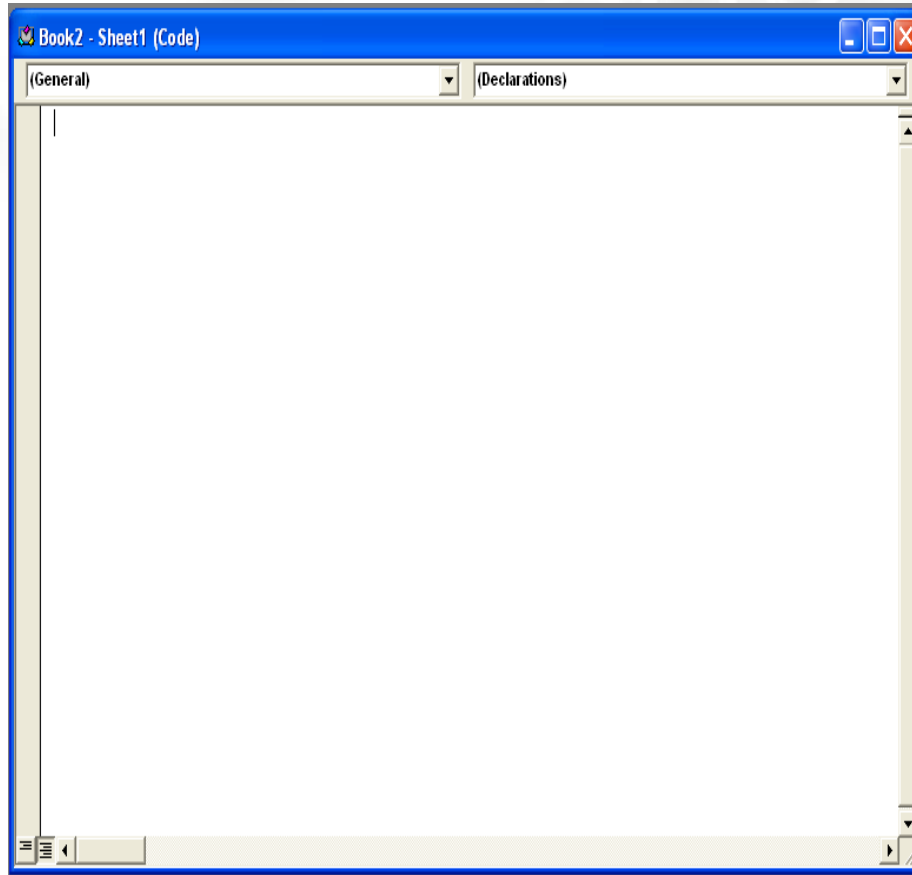
# Setting up the Code Window

- Double click on the name of a component in the Project window (Sheet1, Sheet2, etc.)
- You can also select it via the *View* option on the menu and selecting the *Code* Option.



The Code Window

# The Code Window



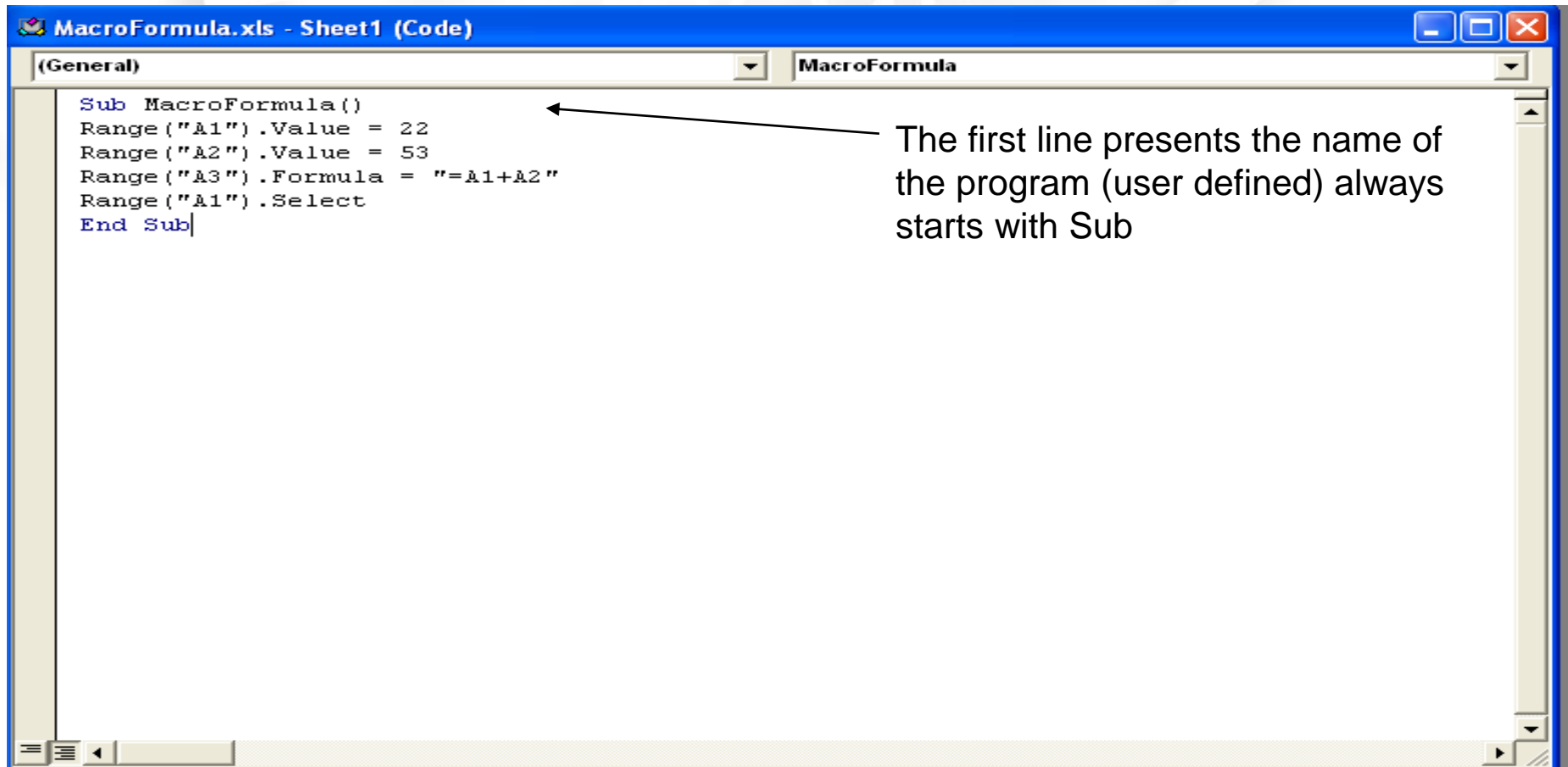
- Macro and VBA code is written and edited here.
- The font style and colour cannot be changed here.
- Text and figures are displayed in black.
- Comments are displayed in **green**, reserved words display in **blue** and error messages appear in **red**.

# Writing a Macro

- Ensure that the Code Window for Sheet 1 is displayed.
- We will enter a simple macro that contains instructions to add together two values and display the total.
- We have to ensure that we enter all text, figures, quotation marks, full stops, brackets, equal signs and spaces accurately.



# Macro Instructions

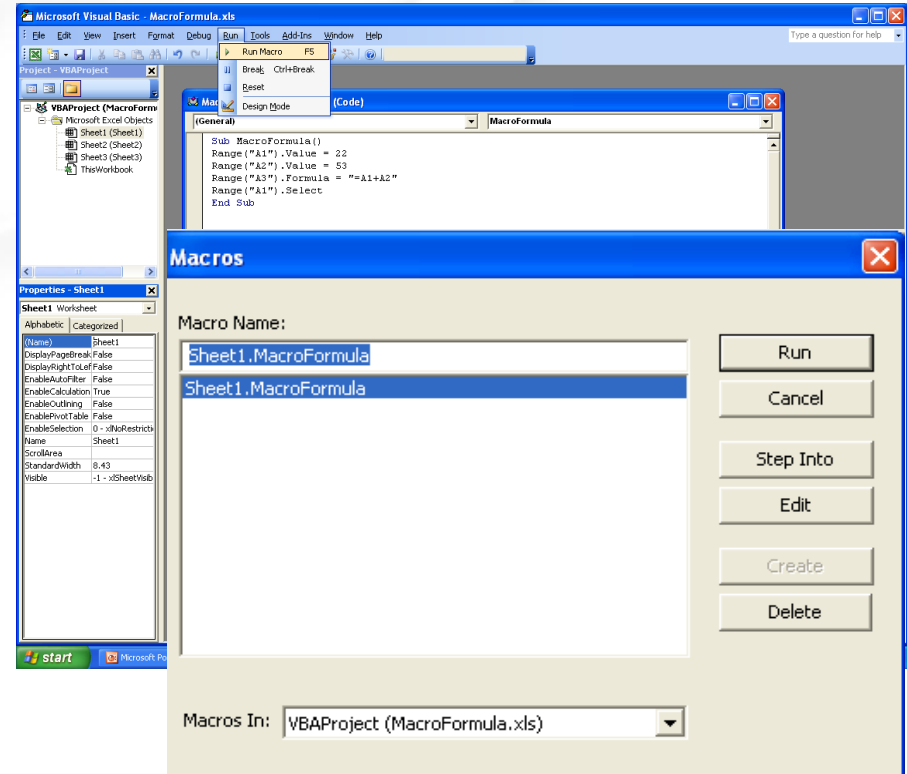


```
Sub MacroFormula()  
Range("A1").Value = 22  
Range("A2").Value = 53  
Range("A3").Formula = "=A1+A2"  
Range("A1").Select  
End Sub
```

The first line presents the name of the program (user defined) always starts with Sub

# Running the Macro

- Select *Run* from the menu bar at the top of the VB Editor screen.
- Select *Run*.
- Select *Run Macro*.
- Run the macro.
- Press *ALT and F11*.



# The Macro Result

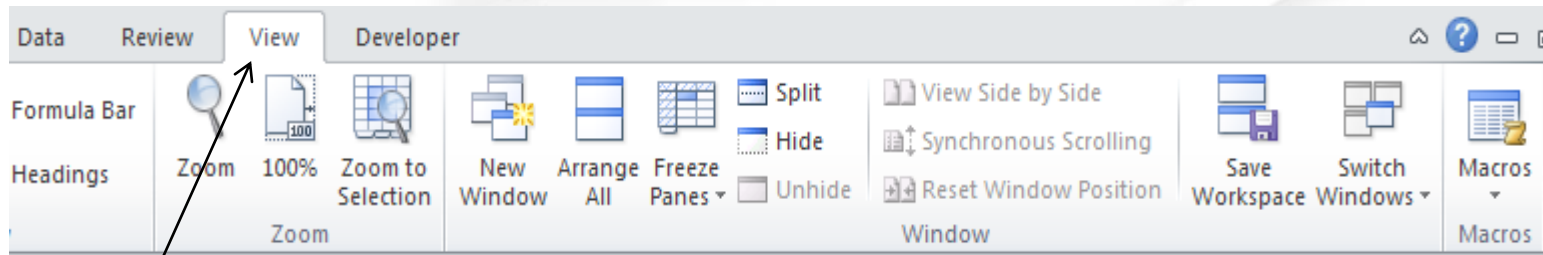
- The two values have been added and the total Calculated.

	A	B	C	D	E	F
1	22					
2	53					
3	75					
4						
5						
6						
7						
8						

# How to Run the Macro from the Menu Bar - 1

- In Sheet 1, delete the figures in cells A1, A2 and A3.
- Select View from the Ribbon.
- Select Macros.
- In the dialogue box, select Sheet1.macroFormula.
- Select Run.

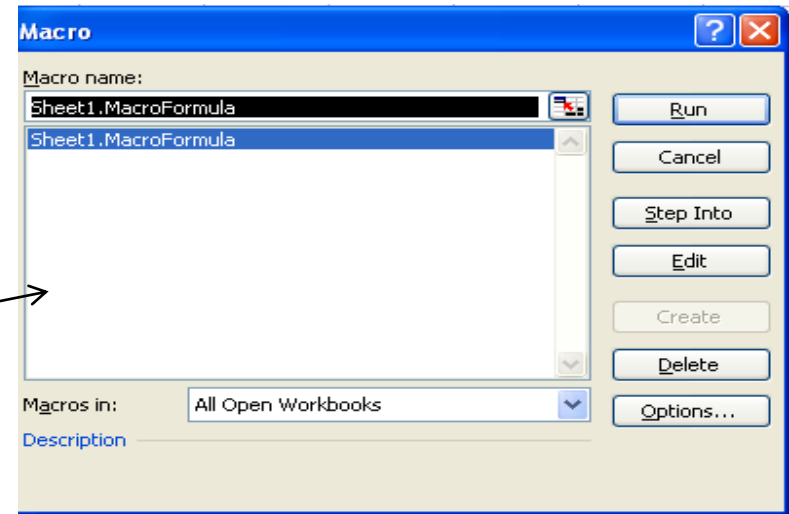
# How to Run the Macro from the Menu Bar - 2



1

2

3



# Result of Running the Macro from the Menu Bar

	A	B	C	D	E	F
1	22					
2	53					
3	75					
4						
5						
6						
7						
8						

# Macro Comments

- A **macro comment** is a piece of text in a macro which will not be executed; it is included to provide the user with information and is displayed in **green**.
- To let the Visual Basic Editor know that it is a comment, place an apostrophe at the start of the text.

e.g.        **'This next instruction will add the two values**  
              `Range("A3").Formula = "=A1+A2"`

- Comments become more useful if programs become more complex and it is good practice to use them.

# Saving a Macro

- This can be done within the VB Editor
- Use relevant names
- There should be no spaces in between words.



# Editing a Macro in the VB Editor

- Select the macro from the list in the Project Window.
- All changes can be made to an existing macro in the VB Editor Code Window.

# References - 1

- Gonzalez, J., Meister, C., Ozgur, S., Dilworth, B., Troy, A. and Brandt, T. (2006). *Office VBA Macros You Can Use Today: Over 100 Amazing Ways to Automate Word, Excel, PowerPoint, Outlook & Access*. Holy Macro Press.  
ISBN-10: 1932802061  
ISBN-13: 978-1932802061
- Jelen, B. and Syrstad. T. (2004). *VBA and Macros for Microsoft Excel*. Pearson QUE.  
ISBN-10: 0789731290  
ISBN-13: 978-0789731296

# References - 2

- Krieger, S. (2011). *Macros Demystified: what they are and why to use them.* [Available Online]  
<http://office.microsoft.com/en-us/help/macros-demystified-what-they-are-and-why-to-use-them-HA010007210.aspx>
- McFedries, P. (2007). *VBA for the 2007 Microsoft Office System.* Pearson QUE.  
ISBN-10: 0789736675  
ISBN-13: 978-0789736673
- Walkenbach J. (2004). *Excel VBA Programming for Dummies.* Wiley Publishing, Inc.  
ISBN: 5000024675

# Topic 5 – An Introduction to VBA and Macros

*Any Questions?*



Bringing British  
Education to You  
[www.nccedu.com](http://www.nccedu.com)

