CSE 123
Introduction to Computing

Lecture 5
Programming with VBA (Macros)

SPRING 2012

Assist. Prof. A. Evren Tugtas

Marmara Üniversitesi

Object Oriented Programming (OOP)

- OOP is an idea of a software consists of distinct objects that have properties and can be manipulated
- These objects exist in the form of bytes and bits.

A bit is a single numeric value → either '0' or '1'

A byte is a sequence of bits → **8 bits = 1 byte**

Visual Basic Applications (VBA)

- VBA: Visual Basic Applications – used by Microsoft (Word, Excel)
- VBA is based on Visual Basic, which is a programming language derived from BASIC
- **BASIC** → **B**eginner's **A**ll-Purpose **S**ymbolic **I**nstructions **C**ode
- Visual basic is *visual* and offers many graphical elements
- VBA consist of Visual Basics implementations that share a common core of objects and commands

Terminology

- **Code:** VBA instructions that are produced in a module sheet when macro is recorded. VBA codes can be manually entered
- **Controls:** Objects on a UserForm that can be manipulated
- **Function:** One or two types of VBA macros that can be created. The other is a Sub procedure
- **Macro:** A set of VBA instructions performed automatically
- **Module:** Contains a VBA code

Terminology

- **Object:** An element that can be manipulated in VBA (Ranges, charts, drawing objects etc.)
- **Procedure:** Another name for macro.
- **Sub procedure:** One of two types of Visual Basic macros that you can create. The other is a function
- **UserForm:** Contains controls for custom dialog box and holds VBA code to manipulate controls
- **VB Editor:** The window used to create VBA macros and UserForms

Macros

- Macros normally used to write short programs
- However, people also use it to write macro viruses
- Excel has two different file extensions as a security against macro viruses;
  - .xlsx – Macro disabled workbook
  - .xlsm – Macro enabled workbook

Macros

Excel takes one of the following four actions when a workbook containin macros is openned;

- Disable all macros without notification
- Disable all macros with notification (default)
- Disable all macros except digitally signed macros
- Enable all macros

To change the default action;

Office/Excel options/Trust Center/Trust Center Settings/Macro Settings

Macros

Trust Center

- Trusted Publishers
- Trusted Locations
- Add-ins
- ActiveX Settings
- Macro Settings
- Message Bar
- External Content
- Privacy Options

Macro Settings

For macros in documents not in a trusted location:

- ( ) Disable all macros without notification
- (•) Disable all macros with notification
- ( ) Disable all macros except digitally signed macros
- ( ) Enable all macros (not recommended; potentially dangerous code can run)

Developer Macro Settings

- [ ] Trust access to the VBA project object model

The Developer Tab

- The developer tab is normally hidden

Excel 2007 users;

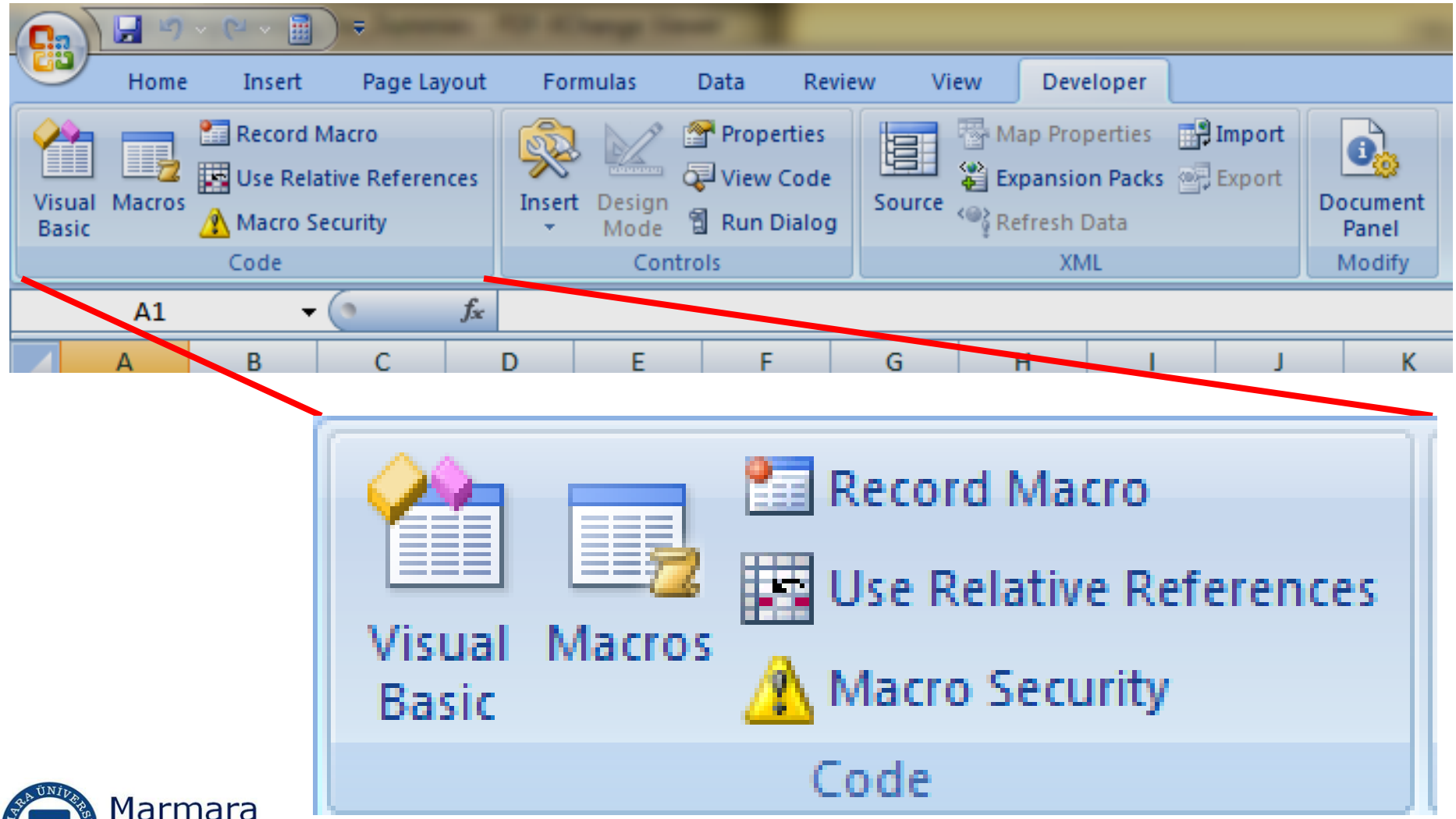
1. Choose File: Excel Options
2. In the Excel Options dialog box, select Popular
3. Place a check mark next to Show Developer tab in the Ribbon
4. Click OK, you will be able to see the Developer tab in your ribbon

The Developer Tab

Excel 2010 users;

1. Rigth click any part of the Ribbon, and choose customize the ribbon
2. In the customize ribbon tab of the Excel Options dialog box, locate Developer in the second column
3. Put a check mark next to the Developer
4. Click OK, you will be able to see the Developer tab in your ribbon

The Developer Tab



Naming the Macro

The name;

- Must start with a letter, after that it can contain both letters and numbers
- Can be up to 80 characters long
- Can contain underscores "_"
- Cannot contain spaces, punctuation, or special characters such as * , !, %, &, ^, / or $
- VBA does not distinguish between uppercase and lowercase letters

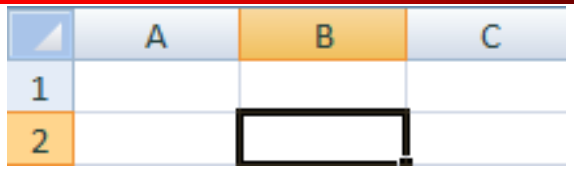
Naming the Macro

- In VBA,
- Any text following a single quote (‘) symbol will be ignored by VBA.
- The single quote indicates that what follows is a *comment* to assist the programmer.

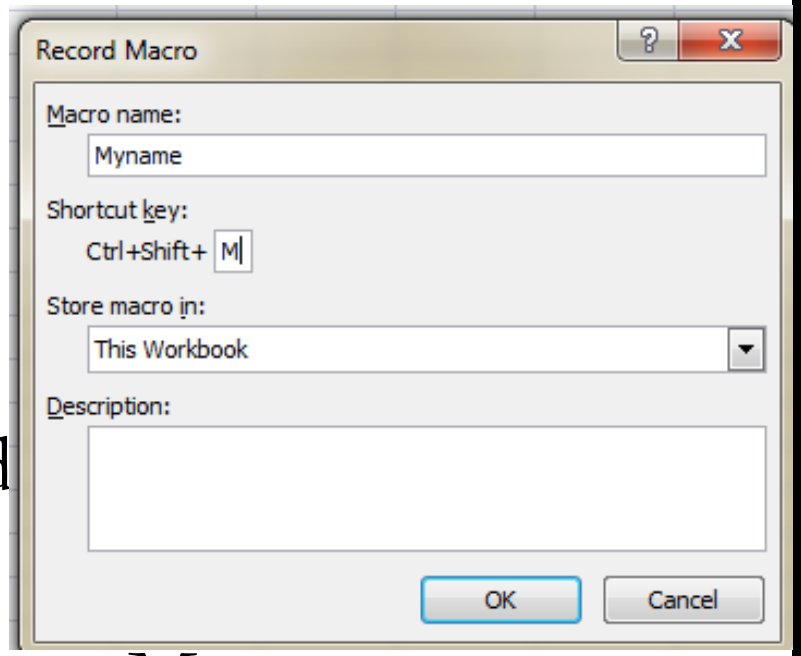
Recording a Macro

- The Excel macro recorder translates your actions into a VBA code.
- When you ask Excel to record a macro, it actually writes a program in VBA.
- Recorded macro can be edited and replayed.
- It is the easiest way to record a macro.

Recording a Macro: A Simple Example



1) Activate an empty cell
2) Choose Developer/Code/Record Macro
3) Give a name (e.g. Myname)
4) You can assing a shortcut key (CTRL+Shift+M)
5) Click OK to record macro
6) Type your name in the selected cell and hit enter
7) Choose Developer/Code/ Stop Macro



Recording a Macro: A Simple Example

- The macro you have recorded will be recorded in a new module named "Module1"
- To view this module
- Choose Developer/Code/Visual Basic

```
Sub Myname()
'
' Myname Macro
'
' Keyboard Shortcut: Ctrl+Shift+M
'
    ActiveCell.FormulaR1C1 = "Evren Tugtas"
    Range("C4").Select
End Sub
```

Comment lines

Recording a Macro: A Simple Example

Microsoft Visual Basic - Book1 - [Module1 (Code)]

File Edit View Insert Format Debug Run Tools Add-Ins Window Help

Ln 10, Col 1

Project - VBAProject

- SOLVER (SOLVER.XLAM)
- VBAProject (Book1)
  - Microsoft Excel Objects
    - Sheet1 (Sheet1)
    - Sheet2 (Sheet2)
    - Sheet3 (Sheet3)
    - ThisWorkbook
  - Modules
    - Module1
- VBAProject (FUNCRES.XL

Properties - Module1

Module1 Module

Alphabetic | Categorized

(Name) Module1

(General)

```
Sub Myname()
'
' Myname Macro
'
' Keyboard Shortcut: Ctrl+Shift+M
'
    ActiveCell.FormulaR1C1 = "Evren Tugtas"
    Range("C4").Select
End Sub
```

Recording a Macro: A Simple Example

- Macro recorded a Sub procedure named Myname

ActiveCell.FormulaR1C1=‘Evren Tugtas’

- This statement causes the name you typed to be written to an active cell

```
Sub Myname()
'
' Myname Macro
'
' Keyboard Shortcut: Ctrl+Shift+M
'
    ActiveCell.FormulaR1C1 = "Evren Tugtas"
    Range("C4").Select
End Sub
```

Testing the Macro

- If you want to replay the macro
- Either use shortcut key (CTRL+Shift+M, in this case)
- Or Choose Developer/Code/Macros/Run

Editing the Macro

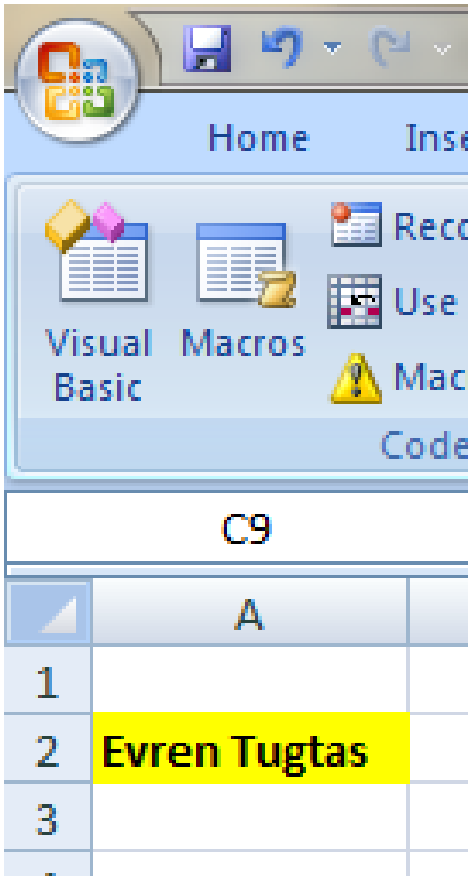
```
Sub Myname()
'
' Myname Macro
'
' Keyboard Shortcut: Ctrl+Shift+M
'
    ActiveCell.FormulaR1C1 = "Evren Tugtas“
    ActiveCell.Font.Bold = True
End Sub
```

Marmara Üniversitesi

Editing the Macro

```
Sub Myname()
'
' Myname Macro
'
' Keyboard Shortcut: Ctrl+Shift+M
'
    ActiveCell.FormulaR1C1 = "Evren Tugtas"
    ActiveCell.Font.Bold = True
    With Selection.Interior
        .Color = 65535
    End With

End Sub
```



Recording a Macro (Temperature)

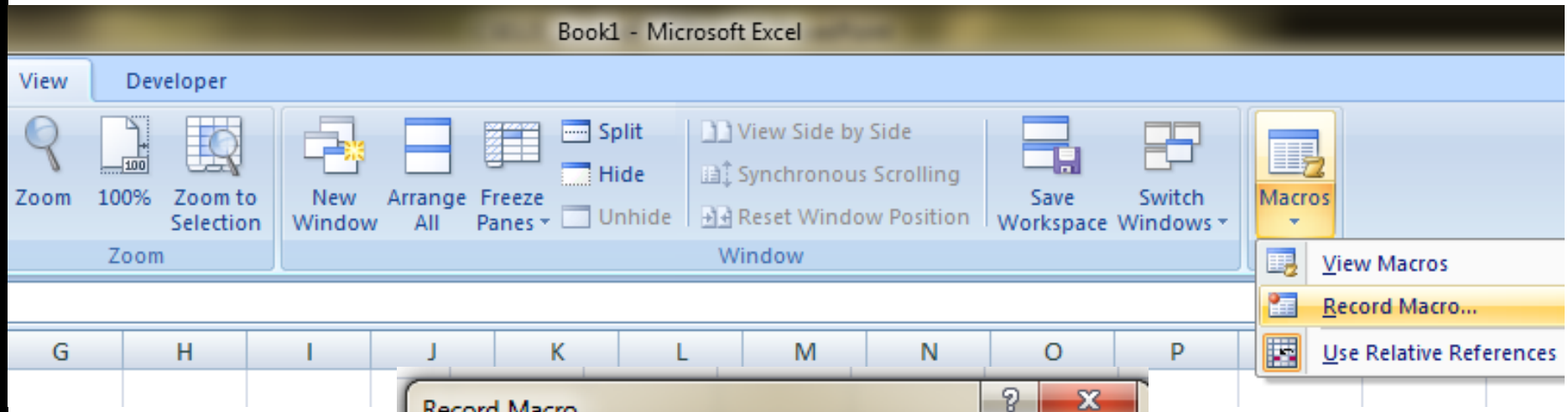
$$T_C = \frac{T_F - 32}{1.8}$$

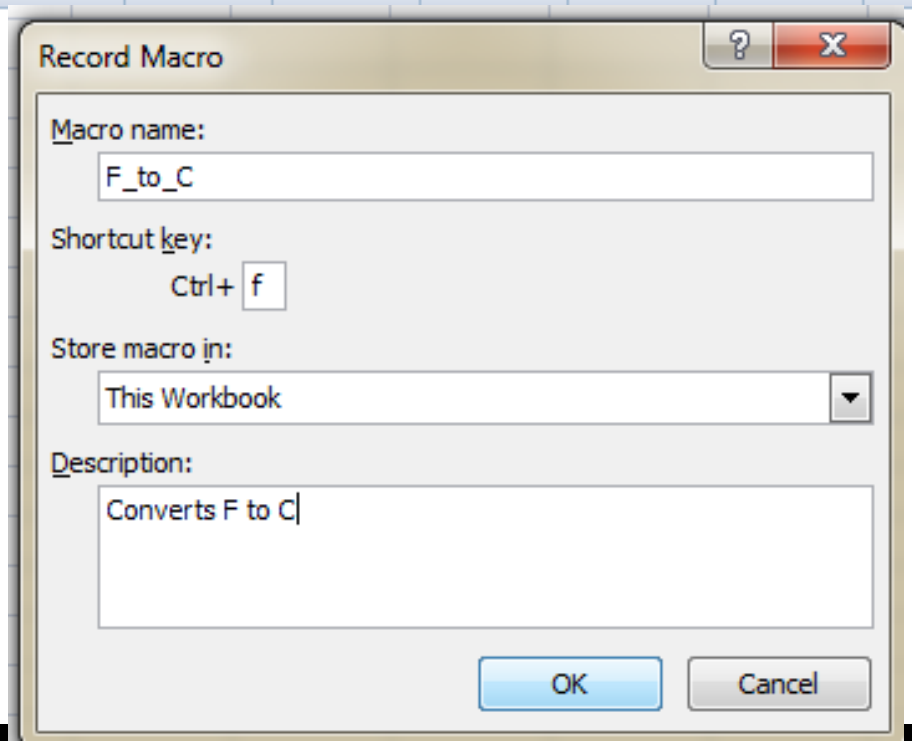
- Record a Macro, which converts temperature in degrees Fahrenheit to degrees Celcius.

D17

	A	B
1	Temp (°F)	Temp (°C)
2	212	
3		
4		

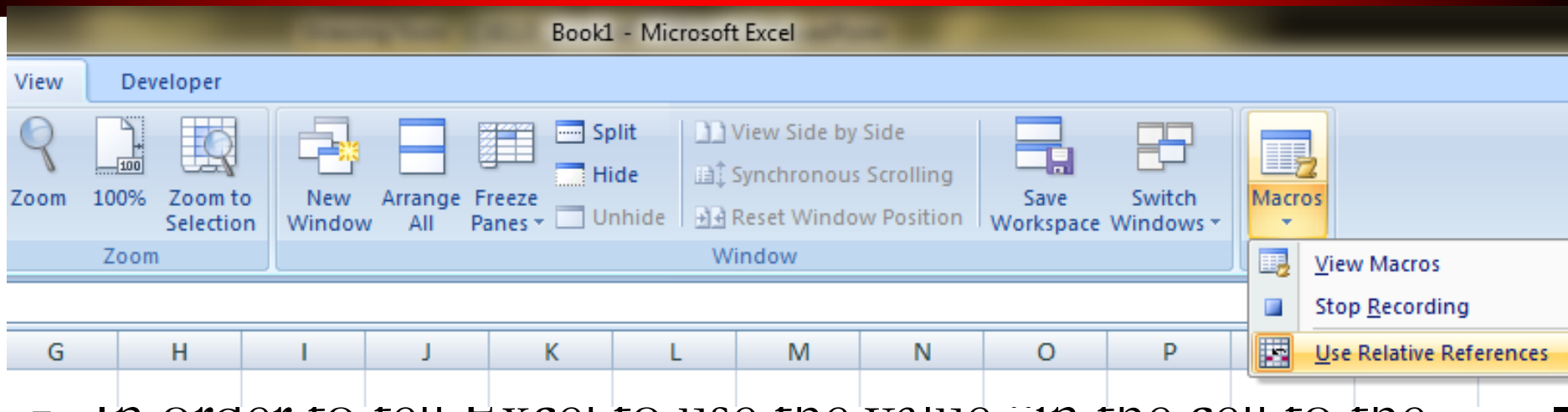
Recording a Macro (Temperature)





Marmara Üniversitesi

Recording a Macro (Temperature)



- In order to tell Excel to use the value "in the cell to the left of currently selected cell" then the macro needs to be recorded using ***relative referencing***
- If you want Excel to use the value in A2 than you need to use ***absolute referencing***

	A	B
1	Temp (°F)	Temp (°C)
2	212	
3		

Recording a Macro (Temperature)

IF =(A2-32)/1.8

	A	B	C	D
1	Temp (°F)	Temp (°C)		
2	212	=(A2-32)/1.8		
3				

- After writting the formula in a selected cell
- You can stop recording the macro

Marmara Üniversitesi

Recording a Macro (Temperature)

```
Sub F_to_C()
'
' F_to_C Macro
' Converts F to C
'
' Keyboard Shortcut: Ctrl+f
'
    ActiveCell.FormulaR1C1 = "=(RC[-1]-32)/1.8"
    ActiveCell.Offset(1, 0).Range("A1").Select
End Sub
```

Recording a Macro (Temperature)

ActiveCell.FormulaR1C1 = "=(RC[-1]-32)/1.8“

R1C1 used for relative cell references

Current row: R one cell to the left of active cell: C[-1]

ActiveCell.Offset(1, 0).Range("A1").Select

Active Cell: Starting from the current active cell location

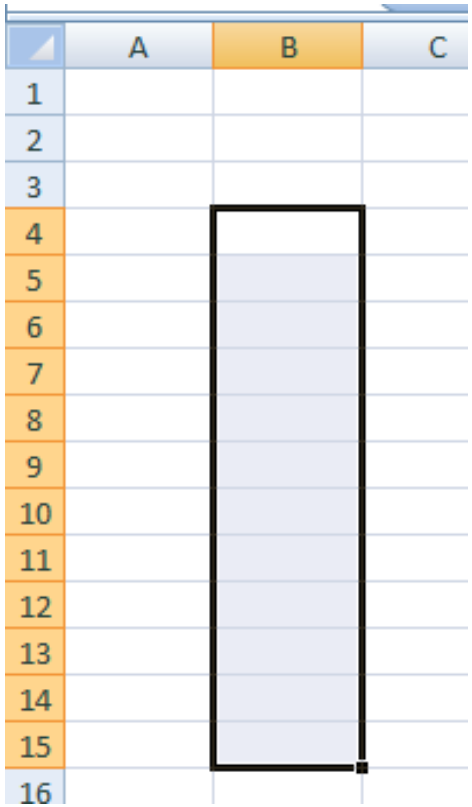
Offset(1,0): Move one row down and zero columns right

Range(“A1”): On currently selected worksheet

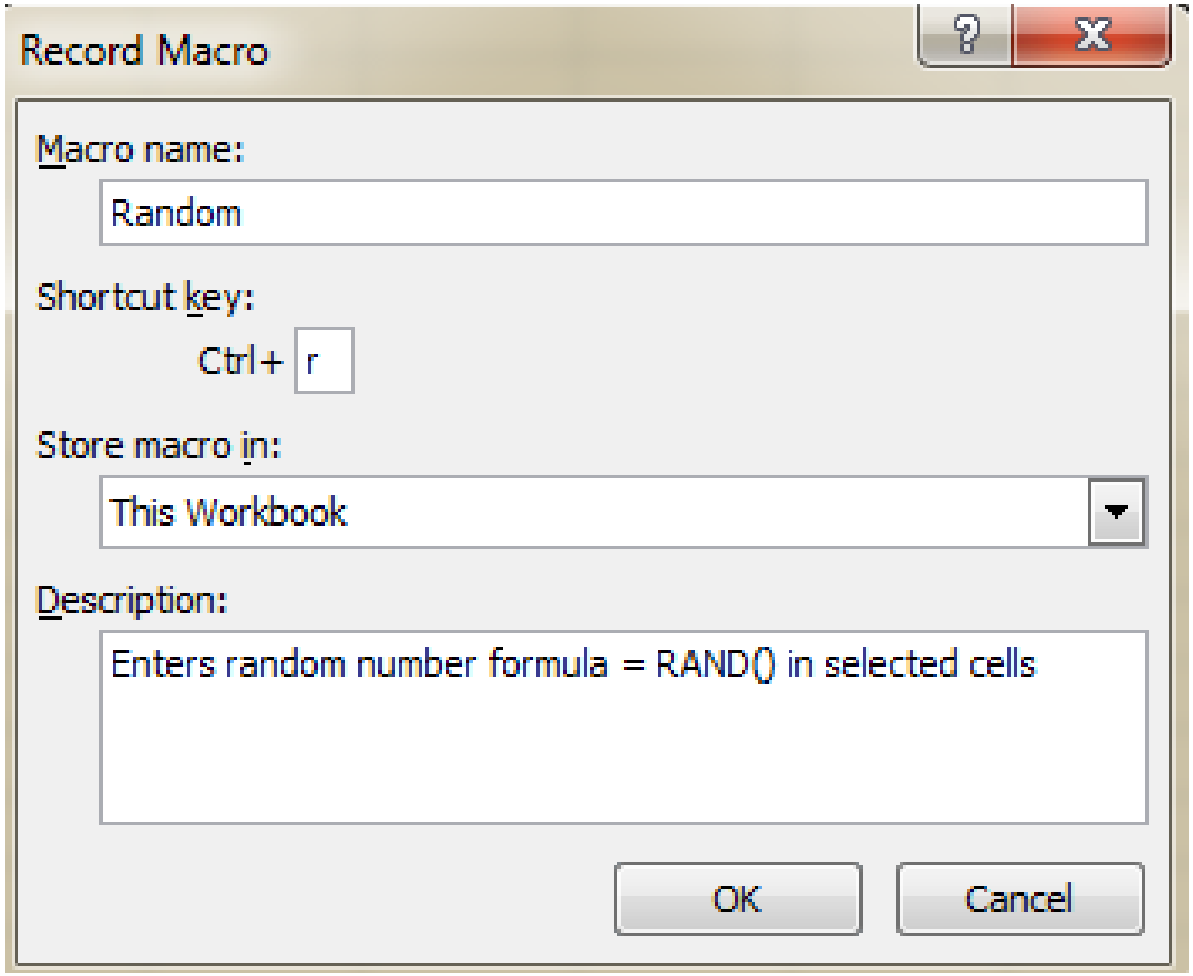
Select : make the offset cell the selected (active) cell

Random Number Macro

- Excels RAND ( ) function returns a random number between 0 and 1
- Select the cells that should contain random numbers
- Record your macro using RAND( ) function



Random Number Macro



Random Number Macro

```
Sub Random()
'
' Random Macro
' Enters random number formula = RAND() in selected cells
'
' Keyboard Shortcut: Ctrl+r
'
    Selection.FormulaArray = "=RAND()"
End Sub
```

Format

- There are predefined constants in VBA to specify line style

xlContinuous

xlDash

xlDashDot

xlDouble

xlLineStyleNone

- To create a solid line: LineStyle:=xlContinuous

Format

- There are predefined constants in VBA to specify line weigth:

xlHairline

xlThin

xlMedium

xlThick

- There are predefined constants in VBA to specify line color:

ColorIndex → Use RGB ( ) function

Format

Table 12.1 ColorIndex and Equivalent RGB Values

	ColorIndex	RGB Values
Black	1	0, 0, 0
White	2	255, 255, 255
Red	3	255, 0, 0
Green	4	0, 255, 0
Blue	5	0, 0, 255
Yellow	6	255, 255, 0
Magenta	7	255, 0, 255
Cyan	8	0, 255, 255

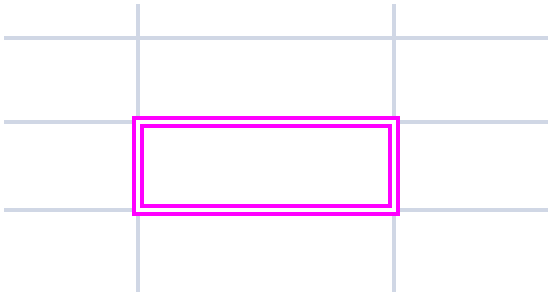
Ref: Larsen, R.W.Engineering with Excel. 3rd ed. New Jersey. Prentice Hall. 2009

Format

Record Macro

Enter the Name

Stop Macro and edit the code as follows



eneral)

```
Sub testformat()
'
' testformat Macro
' Draws a double-line in black around the active cell
'
' Keyboard Shortcut: Ctrl+t
'
ActiveCell.BorderAround LineStyle:=xlDouble, Weight:=xlThick, ColorIndex:=7

End Sub
```