

CHAPTER 9

MACROS

- ✓ Recording Macros
- ✓ Forms Toolbar
- ✓ Using User Forms



Microsoft Office
Excel 2003

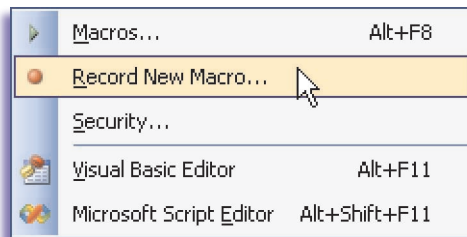


FIGURE 9.1 Tools > Macro menu

MACROS

A macro is a series of commands and functions that are stored in a Microsoft Visual Basic module and can be run whenever you need to perform the task.

If you perform a task repeatedly, you can save the steps as a macro. Anytime you wish to carry out this task, you can simply choose the appropriate macro from the **Macro Menu** on the **Tools** bar.

*In order to run macros, set security to 'Medium' from
Tools > Macro > Security*

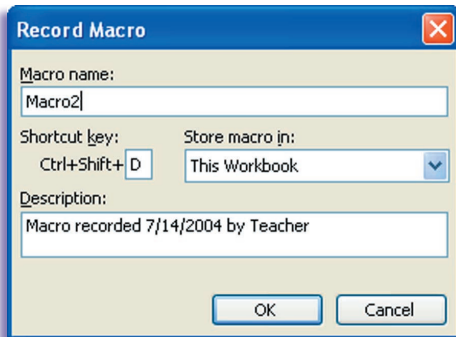


FIGURE 9.2 Record Macro window

9.1 RECORDING MACROS

You can **record a new macro** by opening the **Record New Macro...** window from **Tools** menu **Macro**.

In this window, you can give your macro a name and create a shortcut to your macro. Many programmers forget the details about their programs after some time. It is a good idea to give some brief information about the macro for later use.

After you click the OK button, it will start recording your macro. It will show another small window to stop recording after you finished.

FIGURE 9.3 Stop Recording Macro window

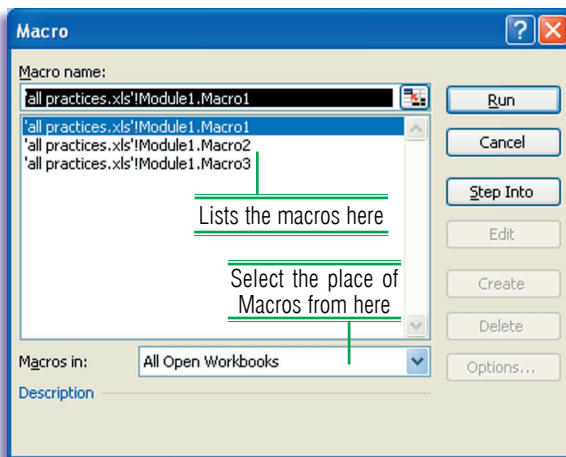


FIGURE 9.4 Load All styles

All of your cell operations and other Excel commands will be stored as a macro until you press the **Stop recording macro** button.

Example 9.1:

Your brother prepares a document for his school homework and in this document he uses the insert and delete cell commands frequently. He wants you to do something to simplify this. ■ Select a range that you want to delete.

- Select **Record New Macro...** command from **Tools** menu
- Write your macro name and give a shortcut for your macro
- Press OK.

Now, it starts recording all your commands. So,

- Right click on the selected area
- Select **Delete** command from the popup menu.
- It will ask you to move the cells up or left.
- Select left and press Enter.

Now you have finished recording macro and you can press the **Stop** button to stop the Record Macro process.

You can see your macro using **Macros...** from the Tools menu. This is the main window for Macros. You can do all operations related to Macros from here. You can Run, Delete, Edit or Create another macro from this window.

When you press **Edit** button, it will open Microsoft Visual Basic Editor.

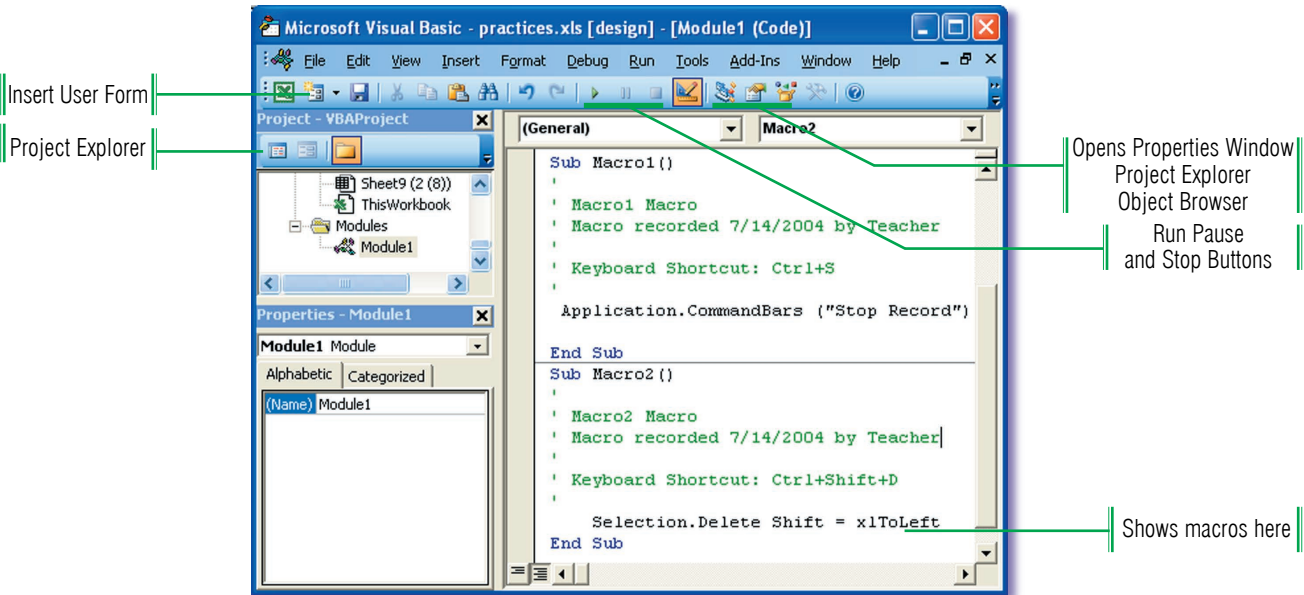


FIGURE 9.5 Visual Basic Editor

9.1.1 Object Browser

Gives a short description and usage information about commands and objects. It shows class names and Members of the selected class. It also gives the properties of the member; the member is either a property or a function of the class.

Now, let's go back to our example. If you edit your macro, it will open MS Visual Basic Editor and show your macro here.

Write your command here to search

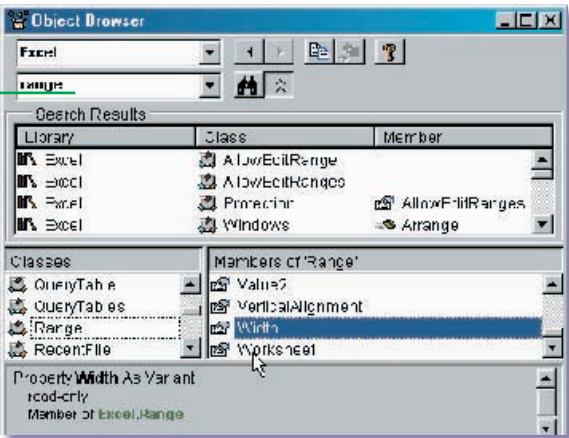


FIGURE 9.6 Object Browser

The header of subprogram	Sub Macro2() ,
Explanation	' Macro2 Macro ' Macro recorded 08.04.2003 by İbrahim Mesecan , ' Keyboard Shortcut: Ctrl+Shift+D ,
The code	Selection.Delete Shift:=xlToLeft End Sub

In the Edit window, you can edit your Visual Basic code or you can rewrite the entire code. Here, most commands from Basic Programming Language, Excel and Visual Basic can be used. The text after the apostrophe is an explanation, not a programming code. **Sub** is a command that says to VB (Visual Basic) that a subprogram with the name Macro2 is starting. And **End Sub** shows that the subprogram has ended. So we have just one line of code here.

```
Sub Macro2()
    Selection.Delete Shift:=xlToLeft
End Sub
```

Now, you finished recording a macro and it is ready to use. Any time you press **<Ctrl+Shift+D>**, it will delete the selected range of cells and move the right cells to left.

9.1.2 ASSIGNING VALUE TO CELLS

Using the **Range** command in Excel, we can define ranges or assign a value to a range.

Range("B2") = 7	'Puts 7 to cell B2
Range("C1:D3") = 5	'Puts 5 to all cells in the range C1:D3
Range("A:A") = 3	'Puts 3 to all cells in column A
Range("5:5") = 4	'Puts 4 to all cells in Row 5
Range("B1") = Range("D7")	'Copies the value of D7 to B1

9.1.3 CONCATENATING TWO STRINGS

Using "&" sign you can concatenate two strings. When you execute the following command

```
Result =Range("C1") & "A"
```

a second string will be added to the end of the first one. So, if **Range("C1")** contains "5" and we add "A" to the end, the value for Result becomes "5A"

Example 9.2:

Your physics teacher heard about your fame at programming from other teachers. He is preparing a multiple choice exam and he wants to check this

exam using Excel. He designed a worksheet similar to the following figure.

He is going to write the answers of a student in this sheet. At the end, he wants to check the answers using a macro. The Macro will take the result and name at the end of a list.

Analyses and Solution

According to your teacher's design:

- The current student's name will be written into C1
- Student answers will be written in to B4:B13
- The cells C4:C13 already contain the correct answers,
- The cells D4:D13 have the formulas to check if the student answer is the same as the correct answer.
- F, G and H columns contain the result list.

Arranging formulas first

Result is a formula that checks if the answer of current student is the same as the correct answer. For the first question, it will check if C4 is equal to B4 ($=C4=B4$). If both are the same then it will give **TRUE**, otherwise **FALSE**. Total correct is another formula that uses the 'Countif' function to count the number of TRUES.

$=\text{COUNTIF}(D4:D13,\text{TRUE})$

Cell E4 will contain the active student order. The name and the result will be copied into that row in columns G and H. In the example above, Todd Williams will be saved into the 7th row in columns F, G and H.

Before running a macro

These are the formulas that will remain throughout all macro runs. After that, for each student, the teacher will write the name into C1 and student's answers into B4 through B13. Then the formulas will give the total correct answers of the current student and it is ready to run the macro.

9.1.3 Record your macro

After entering all data, we will call the macro for this student.

- For the row number of the destination, we will use the number in E4.
- Copy the student name from C1 into destination row in column G
- Copy result from D14 into column H.
- After that increment the number in E4
- And clear old data (The cells B4:B13 and C1)
- Select C1

If you record this macro, it will be something similar to the following code:

FIGURE 9.7 Worksheet design of the Macro

```

Sub Macro1()
'
' Macro1 Macro
' Macro recorded 03/01/2005 by Ibrahim Mesecan
'
' Keyboard Shortcut: Ctrl+Shift+D
'
    Range("C1").Select    'Copies name
    Selection.Copy
    Range("G7").Select
    Selection.PasteSpecial Paste:=xlPasteValues

    Range("D14").Select    'Copies result
    Selection.Copy
    Range("H7").Select
    Selection.PasteSpecial Paste:=xlPasteValues

    Range("E4").Select    'Copies order
    Selection.Copy
    Range("F7").Select
    Selection.PasteSpecial Paste:=xlPasteValues

    Range("E4").Select    'Assigns the new value for order
    ActiveCell.FormulaR1C1 = "8"

    Range("C1").Select    'Clear old name
    Selection.ClearContents
    Range("B4:B13").Select 'Clear old answers
    Selection.ClearContents
    Range("C1 ").Select    'Select name cell
End Sub

```

Record finished but we need to make some modifications to this code. First of all, this macro will always write the results in the 7th row. So, we need to modify the resulting row order. For this purpose we can use a variable for RowOrder. We will take RowOrder from E4 with,

```
RowOrder = Range("E4")
```

Then, instead of writing G7, we can concatenate and form the new storing address as

```
Range("G" & RowOrder)
```

Secondly, we don't have to say copy paste every time. Instead, we can directly assign the source value to destination cell.

```
Range("G" & RowOrder) = Range("C1") 'Copies the name into column G
```

Then, our final program becomes:


```
Sub Macro1()  
,  
' Macro1 Macro  
' Macro recorded 03.01.2005 by Orhan Kararli  
,  
' Keyboard Shortcut: Ctrl+Shift+P  
' & sign is used to concatenate 2 strings  
RowOrder = Range("E4") 'Takes the row order from E4  
Range("G" & RowOrder) = Range("C1") 'Copy the name into column G  
Range("H" & RowOrder) = Range("D14") 'Copy the result  
Range("F" & RowOrder) = RowOrder 'Copy Student order  
Range("E4") = RowOrder + 1 'Increment row order for the next student  
Range("B4:B13") = "" 'Clear the answers  
Range("C1") = "" 'Clear name cell for next student  
  
Range("C1").Select 'select C1 for the next student  
End Sub
```

Example 9.3:

In your school, the science teachers decided to have a general exam every month. It is getting more and more difficult to process data every time, so they decided to use some Macros to quicken the process. They will have a list similar to the following figure.

In data columns, they have class, name information and lessons results. At the end, they want to place the average. The sheet has two main parts. At the bottom part of the sheet, starting from the 6th row, they will place all data. At the top they want to see the results of the query.

	A	B	C	D	E	F	G
1							
2	Class	Name Surname	Math	Phy	Chem	Info	Total
3	9A	John	8	10	10	10	9.5
4	Class Average		7.0	8.5	10.0	9.5	8.75
5							
6	10	Name Surname	Math	Phy	Chem	Info	Total
7	8A	George	7	8	6	8	7.0
8	8B	David	7	7	8	8	7.5
9	8A	Sally	5	9	5	5	6.0
10	9B	Michael	11	5	7	11	5.11
11	9A	John	8	10	10	10	9.5
12	9A	Michael J.	6	7	10	9	8.0
13	8A	Kelly	8	9	7	4	7.0
14	8B	Jill	9	9	7	10	8.8
15	9B	Johnson	8	6	7	7	7.0
16	8B	Kennedy	9	9	7	10	8.8

FIGURE 9.8 Analyzing General Exam Results

The class and name information will be provided into cells A3 and B3. And then, with a shortcut key, the macro will be run. The macro will search all the data for the specified class and student. In the fourth row, they want to see individual lesson averages for the selected class. In the third row, they want specific student’s marks.

Solution

It is quite normal that nobody can know everything. So, for the parts we do not know, we can ask **Record Macro** to show us. Let us say that we do not know how to add a range over another.

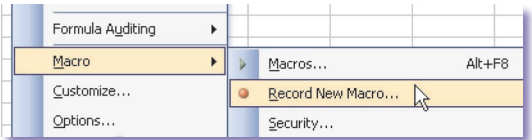


FIGURE 9.9 Running Record New Macro...

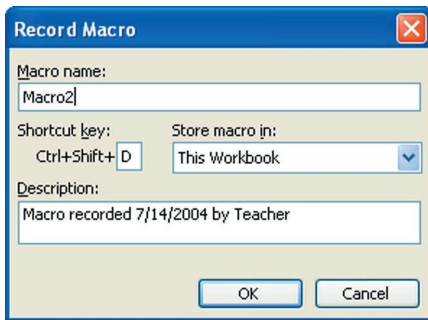


FIGURE 9.10 Record Macro window

When writing shortcuts
be careful not to
overwrite the predefined
(standard) shortcuts.

We run **Record Macro** from **Tools > Macro** menu. It will ask for a macro name and a shortcut. After you press Enter, it will start recording the Macro.

In Record Macro, we want Excel to help us add the value of a cell range onto another. Therefore, we select a range and, from the **Edit** menu we select **Copy**.

After that, we select a destination and call the **Paste Special** command from the **Edit** menu. Finally, from the **Paste special** window, we select

Paste: Value, Operation: Add.

Our macro is ready. Now, we can press the **Stop** button. And from edit macro window, we can see and analyze the code.

```
Sub GetData()  
' Getdata Macro  
' Macro recorded 12/30/2003 by Ibrahim Mesecan  
,  
' Keyboard Shortcut: Ctrl+Shift+D  
Range("C11:F11").Select           'Select source  
Selection.Copy                     'Copy  
Range("C4").Select                 'Select destination  
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlAdd, SkipBlanks :=False, Transpose:=False  
Application.CutCopyMode = False  
End Sub
```

Now you know how to add a range onto another range. In this code, we can skip **Transpose** and **Skipblanks**, because we are not dealing with them. So, we only need to write:

```
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlAdd
```

Now you are ready to write the entire code. To make it easy, your teachers placed the number of students in the A6 cell. First, we get the, Name and Class information, from cells A3 and B3

```
CName = Range("A3") ' Get Class Name from the cell A3  
StName = Range("B3") ' Get Student Name from the cell B3  
LastSt = Range("A6") ' Get The number of Students from the cell A6  
StNum = 0           ' Because, we take the class average, we need to  
                    ' count the number of students in the class
```

```
Range("C4:F4") = "" ' Clear the range for the next operation
```

After we get the initial data and prepare our result part to run, we can start searching from the first until the last student. There is a **For** loop structure in Basic Programming Language which is used to repeat commands or a code.


```

For x = 1 To N
    ' The code here will be repeated N times
Next x

```

Here is the entire code

```

Sub Calculate()
    CName = Range("A3")    ' Get Class Name from the cell A3
    StName = Range("B3")   ' Get Student Name from the cell B3
    LastSt = Range("A6")   ' Get The number of Students from the cell A6
    StNum = 0              ' Because, we take the class average, we need to
                          ' count the number of students in the class

    Range("C4:F4") = ""    ' Clear the range for the next operation

    For x = 1 To LastSt
        ' y is the row order that contains current student name
        y = 6 + x

        ' If Class Name is the same as the current line info then
        If Range("A" & y) = CName Then

            ' Increment the number of students in the class
            StNum = StNum + 1

            ' Copy this line to average part
            Range("C" & y & ":F" & y).Select
            Selection.Copy
            Range("C4").Select

            ' By saying Operation:=xlAdd we take
            ' the sum of each subject when pasting
            Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlAdd

            ' If St Name is also the same as the search info then
            If Range("B" & y) = StName Then
                Range("C3").Select
                Selection.PasteSpecial Paste:=xlPasteValues
            End If
        End If
    Next x

```

```

'Calculate the averages
If StNum <> 0 Then
    Range("C4") = Range("C4") / StNum
    Range("D4") = Range("D4") / StNum
    Range("E4") = Range("E4") / StNum
    Range("F4") = Range("F4") / StNum
End If
Application.CutCopyMode = False
Range("C3").Select
End Sub

```

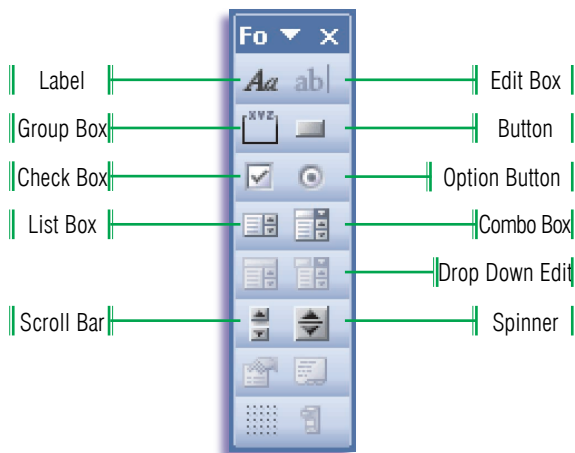


FIGURE 9.11 Forms Toolbar

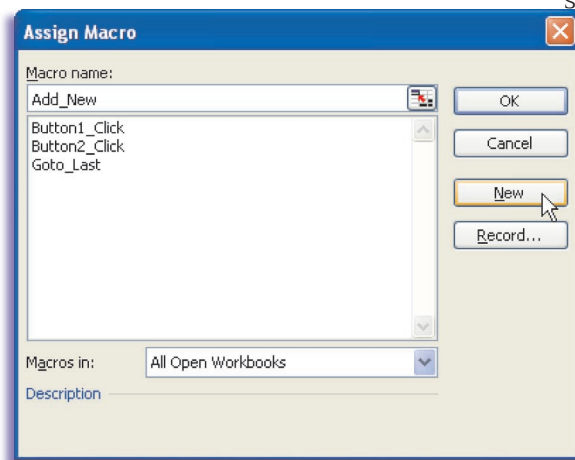


FIGURE 9.12 Assign Macro window

9.2 FORMS TOOLBAR

Until now, we studied how to record a macro; how to modify it; and some main Excel Macro elements. However, there are some other tools which facilitate many operations in Microsoft Excel Macro. The **Forms Toolbar** is one of these tools.

Now we will study the Forms toolbar and use it together with Macro. Writing code is very easy with these buttons. As with all other toolbars, you can open the Forms toolbar from **View>Toolbars>Forms**.

9.2.1 Button

When you select the **Button** tool, It will let you draw the size and place of the button. After you define the place, it automatically opens the **Assign Macro** window and asks you the name of the macro. You can write a new macro name and press the **New** or **Record** buttons. You can also select one of the existing macros from the list to assign.

Any time you need, you can see this macro using the **Macros...** command from the **Tools** menu (or right click on the button and select **Assign Macro**) and edit it as you wish.

9.2.2 ComboBox

The **ComboBox** is used to list some items and allows you to select one of these items. A selected item will appear in the main box. Different than other objects, Forms toolbar

items have another tab in the **Format Control** window: The **Control** tab. Using this tab, you can change the controls like; **Input Range**, **Cell Link** and some other status options for the **ComboBox**.

Input Range defines the items that will appear in the list box. **Cell Link** defines the cell that will have the result of the selection. Using the **Drop down lines** box, you can define the number of lines in the drop down list. Similar to buttons and other **Forms Toolbar** items, you can assign a macro to the drop down ComboBoxes. This macro will be automatically called every time you change the value in the ComboBox. You can use the same methods, which we described for buttons, to assign a macro.

9.2.3 CheckBox

CheckBox is usually used in true false questions. You can find samples of **CheckBox** in many programs. Except the initial state, **CheckBox** has two possible states; **Selected** or **Unselected**. For the initial state, there is a third option: **Mixed** which means **State is not available** (not True but also not False). Same as **ComboBoxes**, from

Format Control > Control Tab, you can assign a cell link and define other status options. The Cell Link will contain a TRUE value if the check box is checked. Otherwise it will be FALSE.

9.2.4 OptionButton

The **OptionButton** is also a common button in many programs. It is used when you have many different options but can select only one. When you have two possible options, it is better to use a **CheckBox**. If you have three or more options and only one of them can be selected at a time, then, it is better to use the **ComboBox** or **OptionButtons** group.

For example, the **Security** option in **Macros** has four possible choices; Very high, High, Medium, and Low. You can select only one of the choices. In most multiple choice exams only one answer can be selected: A, B, C, D or none.

The **OptionButton** has also the **Assign Macro** option and **Format Controls**. In control tab it has **Cell Link**, **Value** and **3-D effect** options.

Example 9.4:

The accountant in your father's company is using Excel worksheets in his balances but he has some difficulties. So, he wants to automate his balance operations. As it is shown below, he designed two main panes. First part is for entering data. Second part is the database that keeps all records and is beneath the data entrance part. At the top, he wants to see the total cash amount.

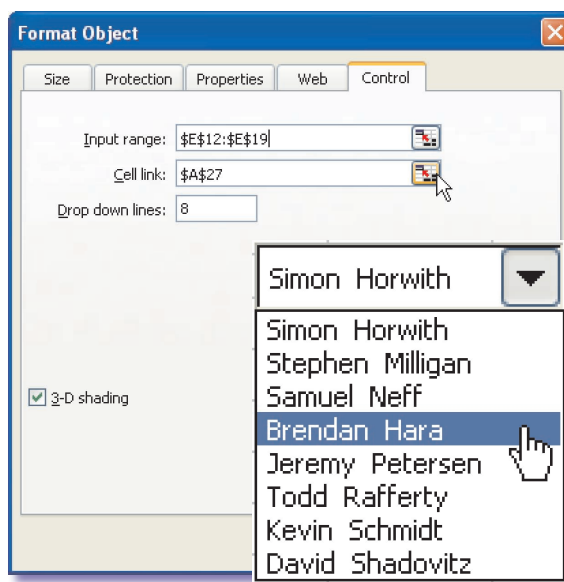


FIGURE 9.13 Format Control window



FIGURE 9.14 ComboBox

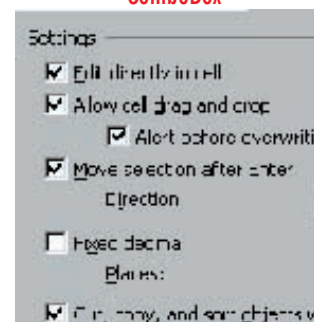


FIGURE 9.15 Some examples of check boxes

For his expenses, he prepared a list of categories and numbered them. Instead of memorizing many categories with their numeric values, he wants to select the category type from a ComboBox, and write the date, explanation and the amount. After pressing a button or a shortcut, he wants to put this record into the database.

Analyses and Solution

First, we design the worksheet as in the figure below. We write our new expenses in row four. We have just one ComboBox and its cell link is B4. The ComboBox input range can also be in another sheet showing the expenses (the table on the right).

	A	B	C	D	E	F	G
1	TOTAL EXPENSES						
2	7475						
3		Expense Type	Date	Explanation	Amount		
4	10	Seminars					
5							
6		Expense Type	Date	Explanation	Amount		
7	9	6	16.May	Seminar in Boston	2500		Expenses
8	8	5	16.May	Seminar in California	380		Salary
9	7	3	16.May	A computer for accountancy	650		Kitchen
10	6	3	16.May	A computer for Secretary	650		Inventory
11	5	1	16.May	John Clayton	1500		Repairing
12	4	4	16.May	100 kg White paint	115		Seminars
13	3	3	15.May	A new table for the big hall	500		Transportation
14							

FIGURE 9.16 Accountancy Balance Sheet

Secondly, we select the expense type from the ComboBox. Since the cell link of the ComboBox is B4, the result will appear at the back of the ComboBox. We will then write the date, explanation and the amount of payment. After we call the Macro, it will copy and insert range A4:E4 to the top of the list, A7:E7. After then, it will clear range C4:E4 and increment the number in A4 for the next operation.

The sum of the numbers will appear in A2. But for this **Sum** formula our starting cell must be less than 7 (6 is OK). When we take seven as a starting point, every time we insert to the top of the list, the starting address of the **Sum** formula will also move down and the new inserted record will not be included in the formula.

Now we know what to do, and all this can be recorded by the Macro recorder. But for incrementation, you need to change the formula,

Range("A4") = Range("A4") + 1

Otherwise, the number you have written will be shown as a direct value and will be repeated every time.

Sub Save()

'

' Save Macro

' Keyboard Shortcut: Ctrl+Shift+S

Range("A4:E4").Select

Selection.Copy

Range("A7:E7").Select

Selection.Insert Shift:=xlDown

Range("A4") = Range("A4") + 1

Range("C4:E4") = ""

End Sub

'Select

'Copy

'Select destination

'insert

'Increment A4

'Clear for the next operation

Example 9.5:

Your math teacher is organizing a contest throughout the school. Since there are many students participating, he does not want to read answer sheets one by one. He wants to use a macro to enter the answers and check them. To simplify the process for you, he does not need to keep the student results in the computer (after you design this level, he can ask once more to upgrade it :)

In the table, he has correct answers in column C. And a formula in column D; giving +8 points, if the answer is correct; zero points, if the answer is left blank; otherwise -1. For now, he wants some option buttons. Every time he clicks on an answer button, that answer will be put into the student's answers table in Column B.

Analyses and solution

After we design our worksheet and insert 5 option buttons, we can select a cell as a cell link (for example C2) of the option buttons. Student's answers will appear in column B, column C will have correct answers and column D will have the result for every question.

When writing the formula for the first question, we will first check if B4=C4 or not. If it is, then the result is 8. Otherwise, we will have another 'If' to check whether cell B4 is empty or not. If B4 is empty then the result is 0, otherwise the result is -1. So, the formula is clear then;

`=IF(C4=B4,8,IF(B4="",0,-1))`

The main program is actually with option buttons. Every time we click on an option button, we want the answer we click to appear in the cell that **Question Number** gives in the cell D1. For example, when **Question Number** in D1 is 11 and we click C, we want the Macro to write C in cell B14 and increment the Question Number to 12.

Sub Answers()
 answer = Range("C2")
 QN = Range("D1") + 3

 If answer = 1 Then Result = "A"
 If answer = 2 Then Result = "B"
 If answer = 3 Then Result = "C"
 If answer = 4 Then Result = "D"
 If answer = 5 Then Result = ""

 Range("B" & QN) = Result

 QN = QN + 1
 Range("D1") = QN + 3

'Option button has been clicked
'Check, which one has been clicked?
'Get current Question Number

'Convert option button to a letter

'Write the result in column B

'Increment QN for next operation and
'store QN

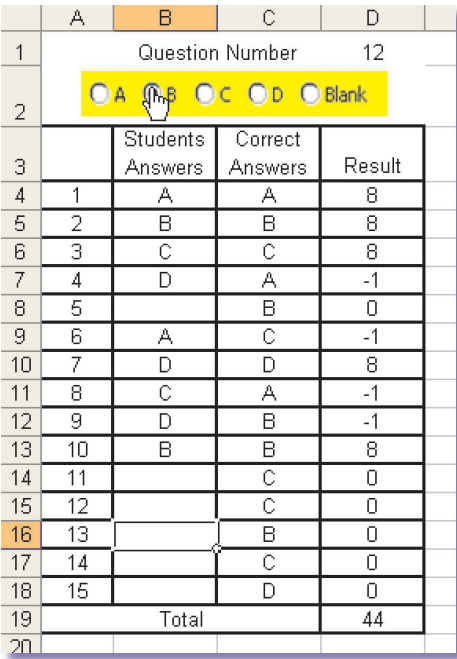


FIGURE 9.17 Multiple Choice Exam Automation

QN = QN - 3

```

If QN > 15 Then                                'After the last question show a MsgBox
    res = Range("D19")
    C=MsgBox("The result is" & res & ", Clear the answers?", vbOKCancel,"Information")
    If C = 1 Then
        Range("D1") = 1
        Range("B4:B19") = ""
    End If
End If
End Sub

```

For the Macro, when we click on an OptionButton, it will read the button from the cell link (C2) and convert it to a letter. After that, it will read the question number, (initially it is 1). But, because we have 3 more lines at the top and our first answer is in the 4th row, we add 3 to **QN** every time and write the result the cell B4 ("B" & QN). For other clicks, it will repeat the same process, until it exceeds the last question. When it exceeds the last question, it is going to show the result and ask whether to clear the old data or not. If you say OK then the old answers will be cleared. Otherwise, you can reset them yourself.

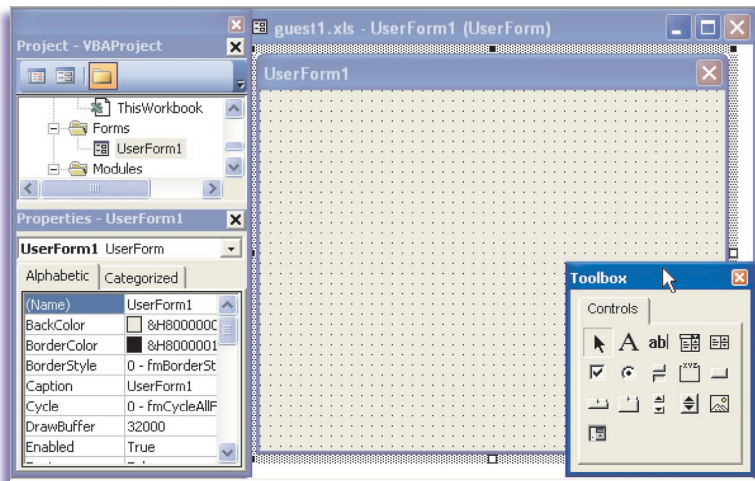


FIGURE 9.18 User Forms

9.3 USER FORMS

Using the Insert **UserForm** button, you can insert **Forms** that have the same properties as the Visual Basic environment.

UserForm contains a nearly similar toolbox. These tools have Visual Basic environment properties, and have some more functions. After you design your Form using controls and arrange their properties, you can show it any time from any macro in the workbook using the command

Userform1.show

Here **Userform1** is the name of the Form and it can be changed from the **Properties**

window. Similarly, in order to hide the User Form you can use the command:

Userform1.hide

Example 9.6:

A nation-wide company wants to prepare a questionnaire for their future product. In the questionnaire there are 20 questions, each having five choices. Because of the high expense the administration decided not to buy

an optical reader for just one questionnaire, and they want you to write a macro that will take all the data into an Excel workbook for analyses.

Your task is to read and concatenate all 20 answers of each interviewer into a cell and all questionnaires into a column. In order to simplify the process, you decided to use option buttons. First you will be asked to enter some specific data; pollster id and number of children in the family. Then, for the other 20 multiple choice questions, every time you click on an option button A B C D or E, it will concatenate the current choice to the end of the answers of the current examinee. After that, it will increment the answer number by one. After the 20th answer, we want Macro to ask whether to save the data or not. If you choose to save, the data will be stored in a special column otherwise the data will be cleared.

Analyses and Solution

Before starting macros, first design the worksheet. You will have 5 option buttons for answers, and a combo box for pollsters. You can have two macros. First macro is Answers() which is assigned to the option buttons. Second macro, Save, will be called from the Answers() macro after processing question number 20. You can select the cell link for Option buttons as E3 and for the Combo box as E4. E2 and E5 are Questionnaire and Question numbers and will be changed from the macros.

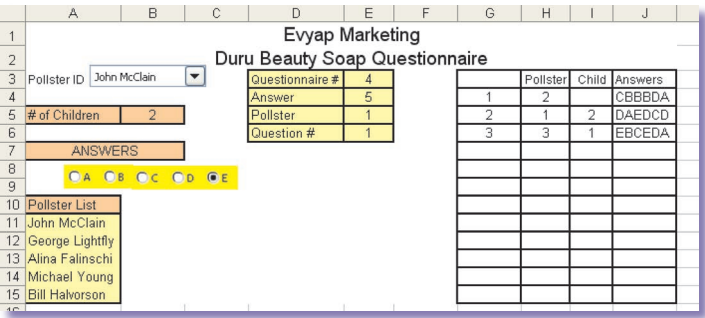


FIGURE 9.19 Questionnaire Automation

The process; first, you are going to select the Pollster and write the number of children into cell B4. Now you are ready to enter the answers. Every time you click on an option button its number will appear in the linked cell E3. So, using E3, you can decide which option button has been clicked.

- If E3 is 1 then converted result will be concatenated "A",
- If E3 is 2 then converted result will be concatenated "B" or
- If E3 is 5 then converted result will be concatenated "E". Then the code can be like,
- If Range("E3") = 1 then Result = Result & "A"

The concatenation operator "&" is used to add the second string to the end of the first one. So, if Result is "DDBC" and we concatenate "A", the new value for result will be "DDBCA".

After we assign the proper letter to the end of the Resulting string, we increment Question number and save it to cell E5. Check, every time, whether the question number's greater than 20 or not. If it is greater, then you call the **Save** sub program; otherwise, we put the new value of Question back to E5.

```

Sub Answers()
    Answer = Range("E3")    'Option button was clicked
    QN = Range("E2") + 2    'Get the Answer for the current question
                              'Get Questionnaire number
                              'QN is for Questionnaire Number

    Question = Range("E5")  'Get Question Number
    Result = Range("J" & QN) 'Get the answers of current examinee

    'Convert the Answer into letter
    If Answer = 1 Then Result = Result & "A"
    If Answer = 2 Then Result = Result & "B"
    If Answer = 3 Then Result = Result & "C"
    If Answer = 4 Then Result = Result & "D"
    If Answer = 5 Then Result = Result & "E"

    Question = Question + 1 'Increment the Question number and
    Range("E5") = Question  'Store back into E5
    Range("J" & QN) = Result 'Store current answers in column J
    If Question > 20 Then Call Save 'If we finish all questions ask for saving
End Sub

```

In Subprogram **Save**, first we ask whether the user wants to save or not. If he clicks the **OK** button, C will get the result 1, otherwise 2. If C is 1 then, we save our data and increment QN by one. Otherwise, we clear our data resetting Question number to 1 and QN will remain the same.

```

Sub Save()
    C = MsgBox("Do you want to Save?", vbOKCancel, "Warning") 'Ask for saving
    Question = Range("E5")    'Get Question Number
    QN = Range("E2") + 2      'Get Questionnaire Number
    Result = Range("J" & QN)  'Get Answers

    If C = 1 Then
        Range("G" & QN) = QN - 2 'You clicked OK and you want to save
        Range("H" & QN) = Range("E4") 'Put Questionnaire number into column G
        Range("I" & QN) = Range("B4") 'Put PollsterID into column H
        Range("J" & QN) = Result 'Put Number of children into column I
        Range("K" & QN) = Result 'Put Answers into column J
        Range("E2") = Range("E2") + 1 'Increment the Questionnaire number
    Else
        Range("H" & QN & ":" & QN) = "" 'You cancel
        'Clear written data
    End If
    Range("E5") = 1 'Reset Question Number
End Sub

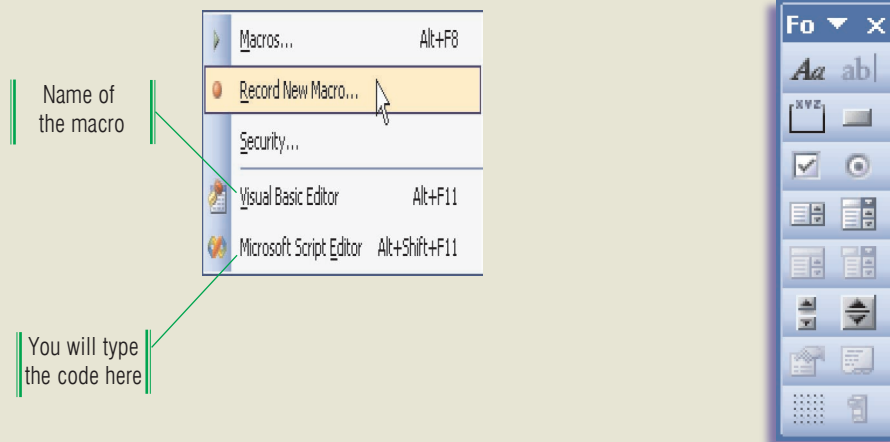
```

CHAPTER 9 in BRIEF

In this chapter, you have learned to automate repeated actions. You can record your doings under a Macro. Anytime you run it, you can repeat your steps with a simple click. You can assign a key combination to your Macro like <Ctrl+shift+4>, and run it via keyboard. Macros can be created using **Record New Macro** from **Macro** on the **Tool** menu. You can also run your the Macro from the same menu. To use macros in your workbooks, the security level **must** be set at medium or low.

You can write your own macros too. All macros are designed in Visual Basic scripting language. After you create the macro, you can see the code of this macro from **Visual Basic Editor**. You also studied how to modify this scripting code. Alternatively, you can manually create your own functions and sub programs by typing directly in this window.

Usually, a Macro starts with "**Sub**" and ends with "**End Sub**". Between these keywords, you type your code. Having finished your code, you can run and see the result.



You can run the Macros assigning it to a button too. You choose the **Forms** from **View, Toolbars** menu. You will find the button tool on this toolbar. On the window just choose your macro's name, and you have assigned your macro to the button. From now on, you can use your macro by clicking this button.

In addition, in the Visual Basic window you can create a **User Form**, which combines the power of Visual Basic code in Microsoft Excel environment. You can view it from any Macro by typing {formname}.show.

QUESTIONS

1. Which of the following statements is not true?
 - A. Excel can use some Visual Basic language commands.
 - B. Macro names can be as long as 32 characters, including letters, numbers and space char.
 - C. You can start the 'Macro Recorder' by selecting 'Tools>Macro>Record New Macro'
 - D. You can assign a shortcut to your Macro to make it quicker to access

2. How can you Run a macro?
 - A. Select Macros from the Tools menu. Then, select your Macro and click the Run button.
 - B. Select Tools-Play Macro from the menu and select macro.
 - C. Click the Play Macro button on the toolbar and select macro.
 - D. Take Music lessons from a professional Macro player.

3. If you perform a task repeatedly in Microsoft Excel, you can automate the task by using a
 - A. Filtering
 - B. Formula
 - C. Macro
 - D. User Form

4. When using Macro, which formula calculates the sum of cells D1 and D2 onto D3?
 - A. $D3 = D1 + D2$
 - B. `Range("D3") = Sum (Range ("D1:D2"))`
 - C. `Range ("D3") = Range ("D2") +Range ("D1")`
 - D. `Range ("D3") = Range ("D2:D3").Copy`

5. Security level must be set to to run a macro in Excel.
 - A. Medium
 - B. High
 - C. Very high
 - D. Ultimate

6. Sub rec()
 Range ("B1:D1").Select
 Selection. Copy
 Range ("D1").Select
 Selection.PasteSpecial Paste:=xlPasteValues,
 Operation:=xlAdd
 End Sub
 If this macro is run, what will the value on cell E1 be?

B	C	D	E	F
2	3	4	6	

- A. 3 B. 4 C. 6 D. 9

7. For the same values in the table in question 6 what will be the value of F1 if you run the macro below?


```
Sub rec1()
  If Range("C1") > 5 Then
    Range("F1") = Range("B1") + Range("D1")
  Else
    Range("F1") = Range("C1") + Range("D1")
  End If
End Sub
```

- A. 6 B. 7 C. 8 D. 10

8. What does the following command do?


```
Range("C4:F4")=""
```

 - A. Assigns 0 value to the range
 - B. Copies '' characters onto the range
 - C. Clears the contents in specified range
 - D. Selects the range

9. From the forms toolbar which option is preferred when you need to use a true/false question?
 - A. Button
 - B. Check box
 - C. Option button
 - D. Label

PRACTICE

- 1- Mr. John Liar, Accountant of MyCowSoft Company, is having difficulty with lots of calculations. He asked for some help from the boss. And the boss has selected you for the job. To help you, Mr. John numbered the expenses. So, in a data board, he keeps the list starting from the 11th row. Time to time, he wants to calculate the sum of each expense type to be written to the summary list at the top.

	A	B	C	D	E
1				TOTAL EXPENSES	
2		Expense Type	Total Amount		
3	1	Salary	14760		
4	2	Kitchen	775		
5	3	Inventory	5280		
6	4	Repair	0		
7	5	Seminars	380		
8	6	Transportation	8200		
9					
10		Expense Type	Date	Explanation	Amount
11	9	6	21 Apr	Michael Young- Seminar in Boston	2500
12	8	5	19 Apr	Michael Young- Seminar Participation fee	380
13	7	3	17 Apr	Computer for Secretary	1280
14	6	1	15 Apr	Staff Salary	7380
15	5	2	13 Apr	Dinner for 25 guests from London	275
16	4	1	11 Apr	Staff Salary	7380
17	3	6	09 Apr	Trip to Bucharest with staff	5700
18	2	2	07 Apr	Lunch with sales representatives	500
19	1	3	05 Apr	Sony Prof. Video Camera	4000

- 2- Champions League Matches: UEFA have fired their computer programmer, because of some disagreements. Now they need someone who can solve their programming problems.

Basically, they have difficulty with their current pointing system and they want to calculate points of each team directly from the scoreboard. In a data board, in an Excel sheet, they store the scores of each match. They want you to write a macro program that will take the data from the board, calculate the points and sort the teams.

	A	B	C	D	E	F
1		Teams	Points	Ordered List		
2		Besiktas		1		
3		Chelsea		2		
4		Spartak Moscow		3		
5		Real Madrid		4		
6						
7		Besiktas		Chelsea		
8		Spartak Moscow		Real Madrid		
9		Spartak Moscow		Besiktas		
10		Real Madrid		Chelsea		
11		Besiktas		Real Madrid		
12		Spartak Moscow		Chelsea		
13		Chelsea		Besiktas		
14		Real Madrid		Spartak Moscow		
15		Besiktas		Spartak Moscow		
16		Chelsea		Real Madrid		
17		Real Madrid		Besiktas		
18		Chelsea		Spartak Moscow		

- 3-Now, you are ready to write the previous project that you prepared for your science teachers. This time they ask for some combo boxes to simplify the selections.

Your teachers will keep the class names in a separate range of cells. There will be two separate combo boxes; one for class names and another for students in the selected class. Any time they select a class from the combo box, the second combo box items will be updated and will have the student list in the new class. When they select any student from the student combo box, it will show the information on the selected student and averages for the selected class.

	A	B	C	D	E	F	G	H	I	J
1		88				Todd Rafferty			4	1
2										
3		Class	Name Surname	Math	Phy	Chem	Info	Total		
4	8B	Todd Rafferty	9	9	7	10	8.8		8A	Jeremy Petersen
5		Class Average	8.3	8.3	7.3	9.3	8.333333333		8B	Todd Rafferty
6									9A	
7			Name Surname	Math	Phy	Chem	Info	Total		
8	8A	Shlomy Gantz	7	8	5	8	7.0			
9	8B	Paul Hastings	7	7	8	8	7.5			
10	8A	Simon Horwith	5	9	5	5	6.0			
11	9B	Stephen Milligan	3	5	7	8	5.8			
12	9A	Samuel Neff	8	10	10	10	9.5			
13	9A	Brendan Hara	6	7	10	9	8.0			
14	8A	Jeremy Petersen	8	9	7	4	7.0			
15	8B	Todd Rafferty	9	9	7	10	8.8			
16	9B	Pete Thomas	8	6	7	7	7.0			
17	8B	Nicholas Tunney	9	9	7	10	8.8			

- 4- Prepare a calculator in Excel.

As a second project, you can also upgrade this calculator to make complex calculations using parenthesis.

Note: Use Macros, do not use the easier method in which you calculate the results without programming.

	A	B
1		0
2		0
3		
4		

+

1

2

3

-

4

5

6

*

7

8

9

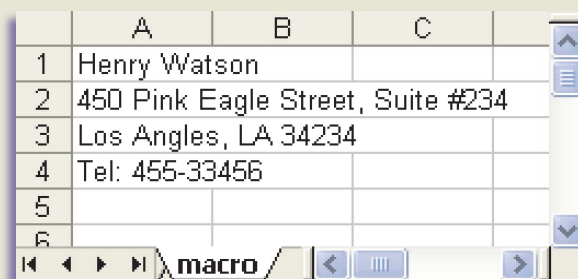
/

0

=

HOMEWORK

1. Create a new workbook and save it as "Macro-1.xls"
2. Select 'Tools > Macro > Record New Macro from the menu.
3. In the Macro name box type "Enter_Address,"
4. Click OK to start recording.
5. Type your name, address, and phone number in the following format:



	A	B	C
1	Henry Watson		
2	450 Pink Eagle Street, Suite #234		
3	Los Angeles, LA 34234		
4	Tel: 455-33456		
5			
6			

6. Click the cell that contains your name and make it bold.
7. Click the Stop button.
8. Clear the address information you just entered and try running your macro. You can modify this Macro to put your signature at the end of your documents.

WORD SEARCH PUZZLE

Find the phrase by filling in the blanks below.

A

RCAMO

SI

A

SSIEER

FO

MMNADSCO

AHTT

SOMATEUAT

EHT A *TOPLAM 3 KARE*

S SKTA *TOPLAM 5 KARE*

CHAPTER 1	CHAPTER 2	CHAPTER 3	CHAPTER 4	CHAPTER 5
1- D	1- C	1-B 16-D	1- D	1-B 16-C
2- C	2- B	2-D 17-A	2- A	2-D
3- B	3- D	3-A	3- -	3-A
4- B	4- B	4-B	4- B	4-B
5- A	5- B	5-D	5- B	5-C
6- C	6- C	6-C	6- C	6-D
7- C	7- A	7-B	7- D	7-D
8- A	8- D	8-A	8- A	8-D
9- D	9- C	9-C	9- A	9-D
10- A	10- FALSE	10-D	10- C	10-C
	11- D	11-C		11-C
	12- D	12-C		12-T
	13- B	13-B		13--
	14- A	14-C		14-D
		15-D		15-D
CHAPTER 6	CHAPTER 7	CHAPTER 8	CHAPTER 9	
1-A	1- C	1-B	1-C	
2-D	2-D	2-B	2-A	
3-C	3-A	3-D	3-A	
4-A	4-B	4-C	4-D	
5-D	5-A	5--	5-A	
6-C	6-A	6-A	6--	
7-B	7-D	7-B	7-B	
8-D	8-TRUE	8-TRUE	8-C	
9-D	9-A	9-FALSE	9-B	
10-B	10-B	10-NOT D		
		11-FALSE		
		12-TRUE		