

MACROS FOR THE LATIN CLASSROOM

(WORD 2003)

MACROS

What is a macro?

A macro is a **mini-program** that executes some **series of steps** without requiring you to manually perform each one. For example, you might frequently make text bold, all caps, 14 pt, and centered. This requires four separate steps, but you can create a macro that will execute all of these steps for you. When you create a macro, you must also specify a shortcut key or a toolbar button to tell the macro to run itself.

How can I create a macro?

There are two ways to create a macro:

1. **“Record” the macro.** Use the instructions on the pages that follow to record a new macro. Essentially, what you will be doing is this:
 - a. Command the computer to memorize every step you take.
 - b. Assign a shortcut key or create a toolbar button that will cause the steps to be executed.
 - c. Perform all the steps you want the computer to remember.
 - d. Indicate that you are finished performing the steps to be remembered.
2. **Write the macro** yourself using the Visual Basic programming language.
 - a. The best way to experiment with Visual Basic is to record a macro then inspect the code in the Visual Basic Editor (hit Alt-F11) to see what commands are used to perform various actions.

How can I create the shortcut keys and toolbar buttons that execute the macro?

You are given the chance to create a shortcut key or a toolbar button when you record the macro, but it is also possible to add shortcut keys or toolbar buttons after the macro has been created.

1. Creating Shortcut Keys:
 - a. Click **Tools** menu then select **Customize**.
 - b. Click **“Keyboard.”**
 - c. In the **“Categories”** section, scroll down and click on **“Macros.”**
 - d. In the **“Commands”** section, click on the macro whose shortcut key you wish to edit, then make the desired changes using the instructions in the pages that follow.
2. Creating Toolbar Buttons:
 - a. Click **Tools** menu then select **Customize**.
 - b. There are three tabs: Toolbars, Commands, and Options. Click on **“Commands.”**
 - c. In the **“Categories”** section, scroll down to **“Macros.”**
 - d. In the **“Commands”** section, click on the macro for which you wish to create a toolbar button.
 - e. Click on the name of the macro and drag and drop it onto the toolbar, then edit the appearance of the button using the instructions in the pages that follow.

MACROS FOR THE LATIN CLASSROOM

(WORD 2003)

How can I edit already-existing shortcut keys and toolbar buttons?

1. Modifying Shortcut Keys: (same steps as adding a new shortcut key)
 - a. Click **Tools** menu then select **Customize**.
 - b. Click "**Keyboard**"
 - c. In the "**Categories**" section, scroll down and click on "**Macros**."
 - d. In the "**Commands**" section, click on the macro whose shortcut key you wish to edit, then make the desired changes using the instructions in the pages that follow.
2. Modifying Toolbar Buttons:
 - a. **Right-Click** on the toolbar button you wish to edit and make the desired changes using the instructions in the pages that follow.

What kinds of things can go wrong with macros?

1. When you record a macro, you may unintentionally ask the computer to memorize steps you don't actually want it to perform. For example...

If you record a macro to change font margins to .5 and .5 on the top and bottom and .8 and .8 on the left and right, the computer will also record other settings such as the orientation of the page (landscape vs. portrait).

If you record a macro to change the font to blue, bold, and italic using the Font dialog box (Ctrl-Shift-F), the computer will also record other settings such as the font name and size.

This problem is fixable! Use the Visual Basic Editor (click Tools Menu - Macro - Visual Basic Editor, or hit Alt-F11) to delete unnecessary lines of code.

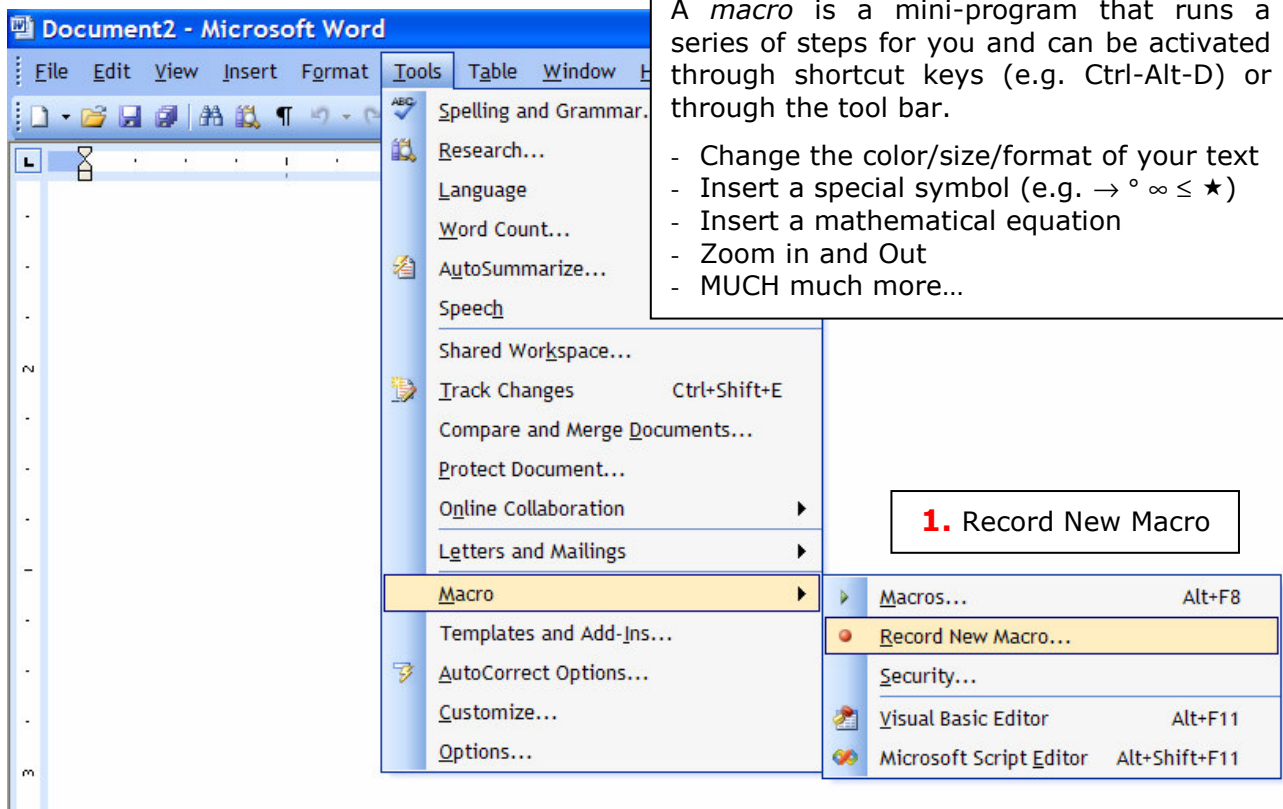
2. If you write or edit macros using the Visual Basic Editor, you may unwittingly create improper commands or delete important lines of code without which the macro will not function properly. Figuring out how to edit and write code simply takes some trial and error and maybe some Google searching to learn more about Visual Basic commands.

Contact Me: *Anna Andresian, Sage Ridge School, Reno NV - anna@magistrula.com*

MACROS FOR THE LATIN CLASSROOM (WORD 2003)

CREATING MACROS

TOOLS MENU - MACRO - RECORD NEW MACRO

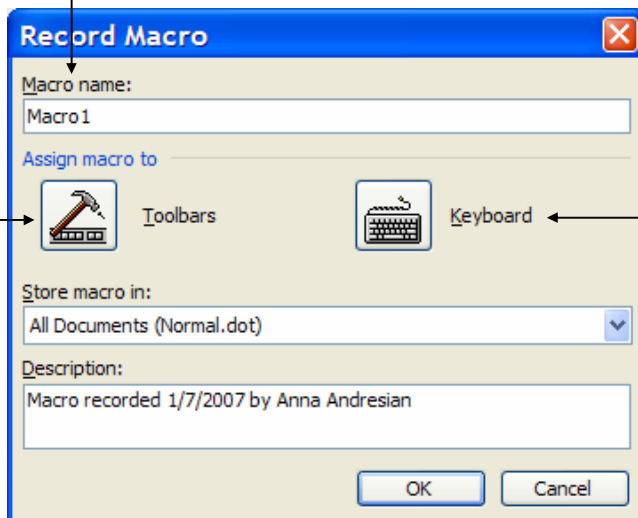


A *macro* is a mini-program that runs a series of steps for you and can be activated through shortcut keys (e.g. Ctrl-Alt-D) or through the tool bar.

- Change the color/size/format of your text
- Insert a special symbol (e.g. $\rightarrow^{\circ} \infty \leq \star$)
- Insert a mathematical equation
- Zoom in and Out
- MUCH much more...

1. Record New Macro

2. Input name of macro:
No spaces or special characters

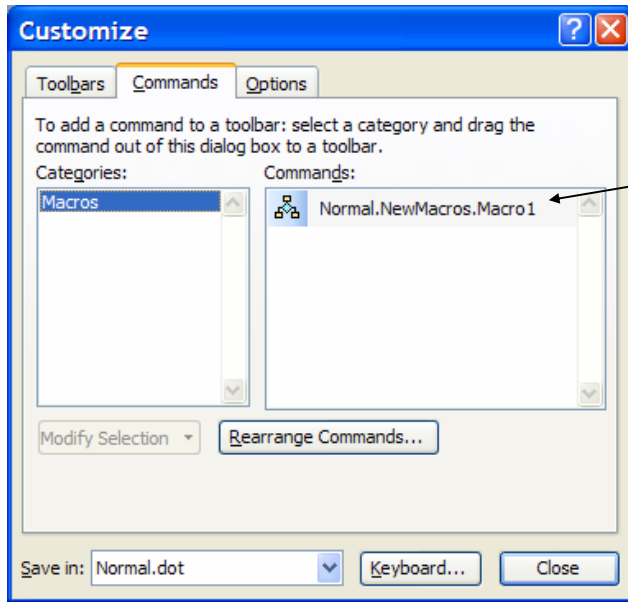


Option A: Press "Toolbars" to proceed to a dialogue box that will allow you to make your macro appear as a button on one of the toolbars.

Option B: Press "Keyboard" to proceed to a dialogue box that will allow you to assign your macro a shortcut key.

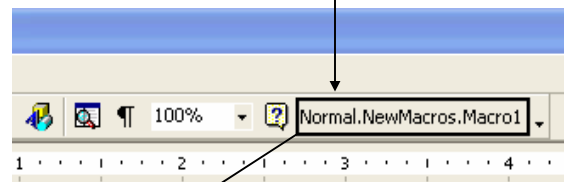
MACROS FOR THE LATIN CLASSROOM (WORD 2003)

ASSIGNING THE MACRO TO A TOOLBAR

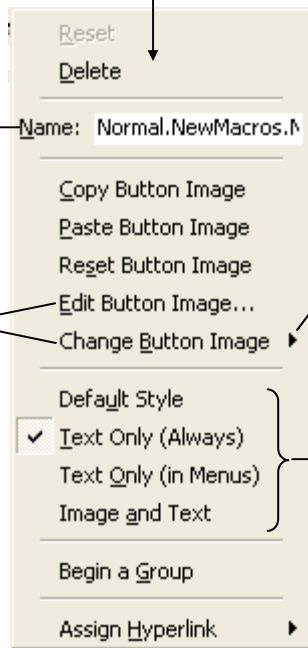


3A. Click here and *hold down* the mouse button. Move the mouse pointer to the spot on the toolbar where you want the button to appear, then let go of the mouse button.

When you drop the macro onto the toolbar, it will look like this.



4A. Right-click on the button to bring up this



5A. Label:
Type the label you want to appear on the button (instead of the default Normal.NewMacro.Your Macro).

6A. Button Image:
You may either edit the button image yourself or click "Change Button Image" to select from a number of preloaded button images.

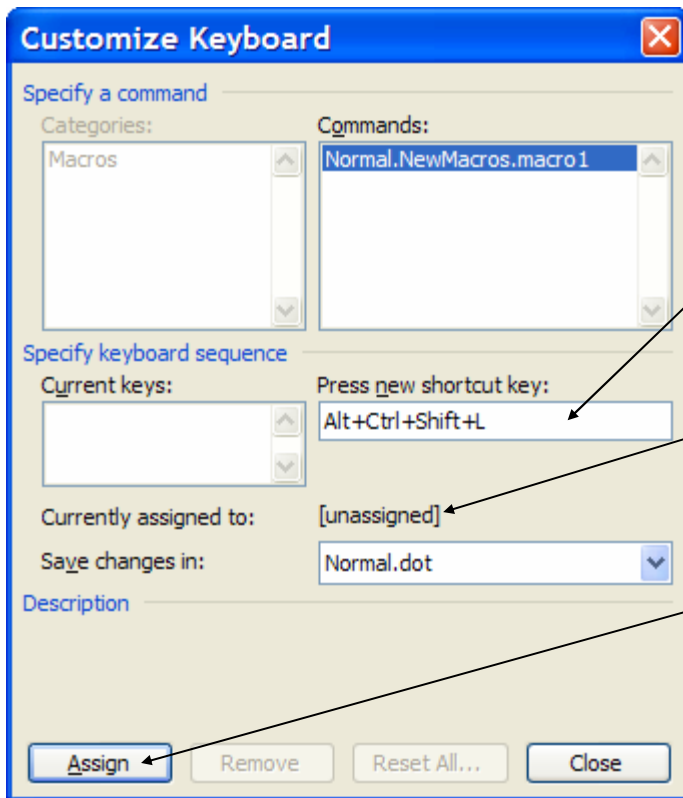


7A. Button Style:
Default: Image only
Text Only: Shows Name
Image & Text: Image & Name

After you have set up the toolbar button, click "Close" to begin recording the macro. Go to **Step 8**.

MACROS FOR THE LATIN CLASSROOM (WORD 2003)

CREATING A SHORTCUT KEY FOR YOUR MACRO



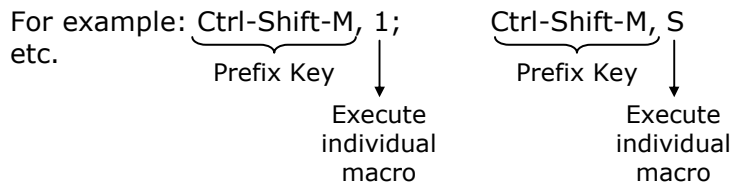
3B. Click in this box then hold down the shortcut keys you wish to use to activate your macro.
(e.g. Alt-5, Ctrl-Alt-T, Ctrl-Shift-M, Ctrl-7)

If your desired shortcut is already assigned to another function, you may:
1) assign it to your macro anyway, or
2) pick a different shortcut key.

4B. When you have chosen your shortcut key, click "Assign."

Prefix Keys: If you find yourself running out of available shortcut keys, Prefix Keys are a great solution.

In the "Press New Shortcut Key" field, hold down a set of keys (this serves as your "prefix") then let go, then hit another key (this will serve to execute your individual macro).



After you have set up the shortcut key, click "Close" to begin recording the macro. Go to **Step 8**.

MACROS FOR THE LATIN CLASSROOM

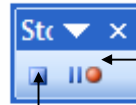
(WORD 2003)

RECORDING YOUR MACRO

8. Record Macro:

Once you have set up a toolbar button or a shortcut key, this box will pop up to indicate that your macro is now being recorded.

It is now time for you to go through the steps you want the computer to take whenever you click your new button or push your shortcut keys.



Click here to pause or resume recording if needed.

9. Stop Recording:

Click here to stop recording once you have performed your steps.

Sample Macro Steps:

View → Zoom → Whole Page

Format → Font → Size = 20, Color = Blue, Small Caps

Insert → Page Numbers → Bottom of the Page, Center

Insert → File → Browse for File

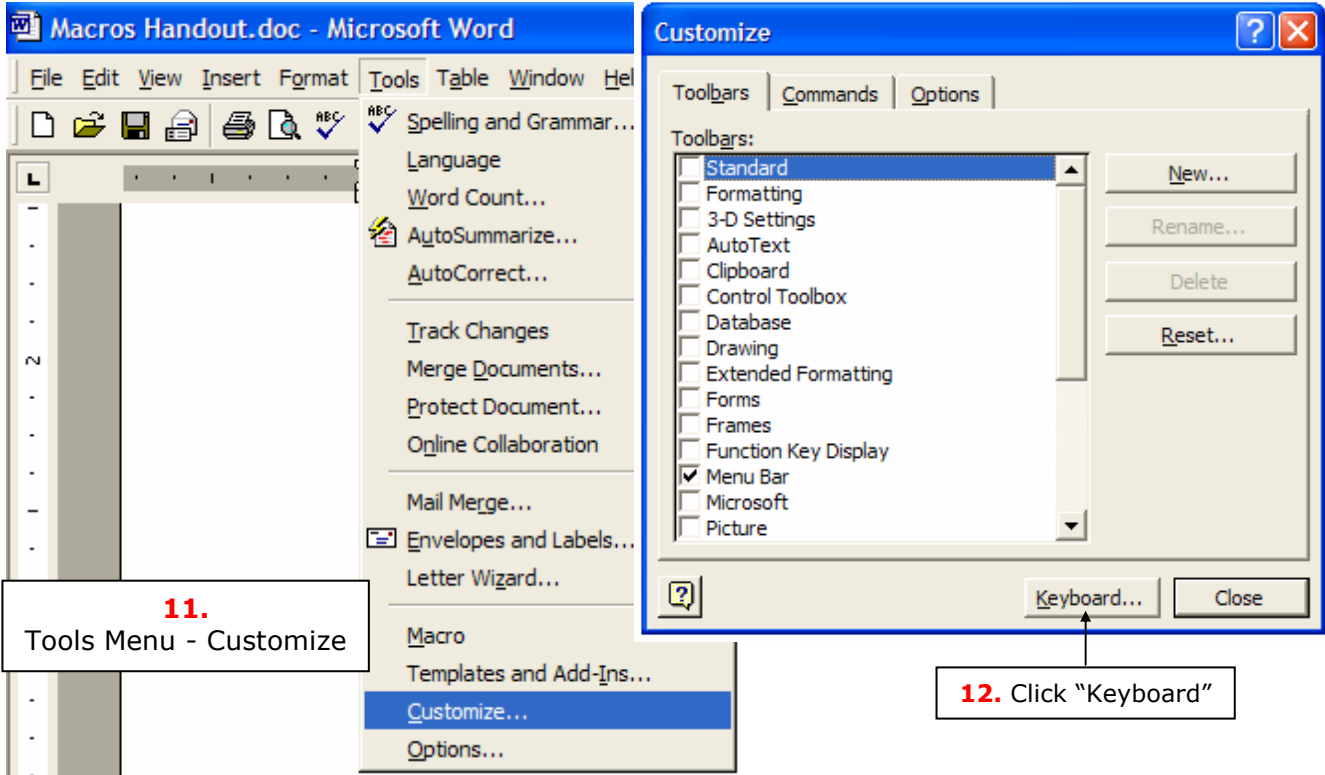
ACTIVATING YOUR MACRO

10. Run Macro: Simply push your new toolbar button or press your shortcut key!

MACROS FOR THE LATIN CLASSROOM (WORD 2003)

MANAGING SHORTCUT KEYS

If you forget or want to change your shortcut key...



11.
Tools Menu - Customize

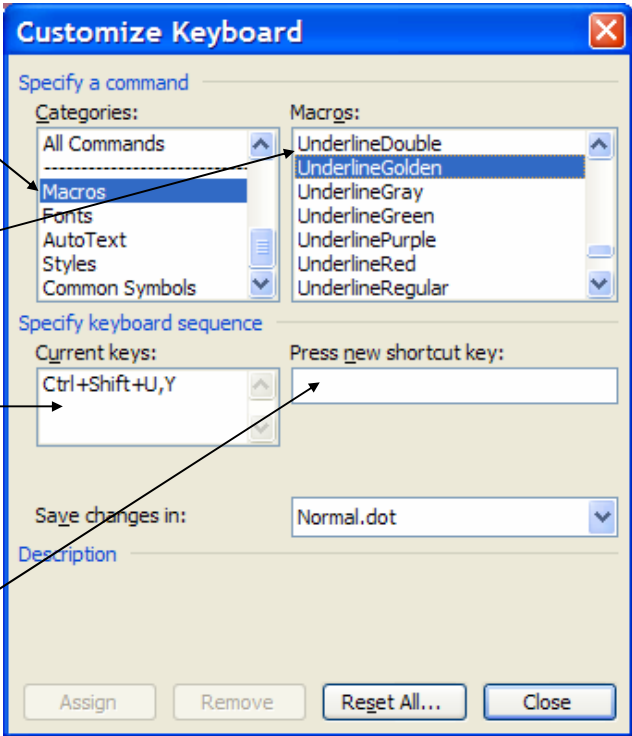
12. Click "Keyboard"

13. Highlight "Macros" on the "Categories" list.

14. Highlight the macro whose shortcut key you wish to view or change.

15. Remove Key:
View existing macros here. Highlight one and click "Remove" if you no longer wish to use that shortcut key.

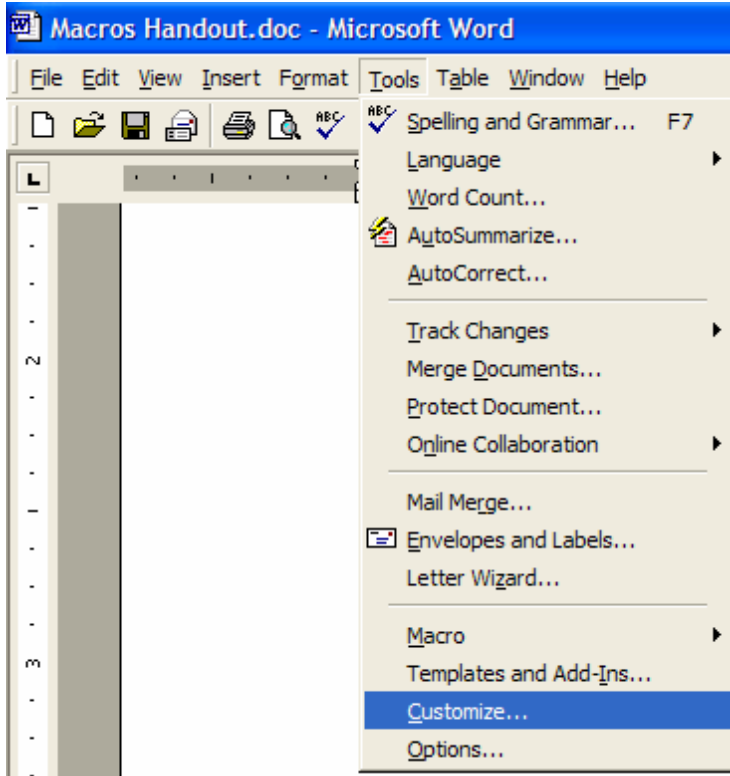
16: Assign New Key:
To add a new shortcut key, click in this box and press the desired keys, then click "Assign."



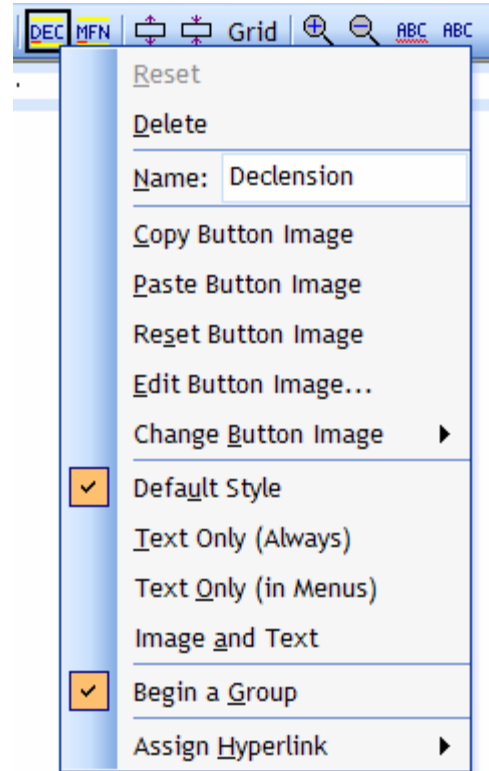
MACROS FOR THE LATIN CLASSROOM (WORD 2003)

MANAGING TOOLBAR BUTTONS

If you want to change your toolbar button...



17. Tools Menu - Customize



18. Right-Click on the button you wish to edit, and make the desired changes.

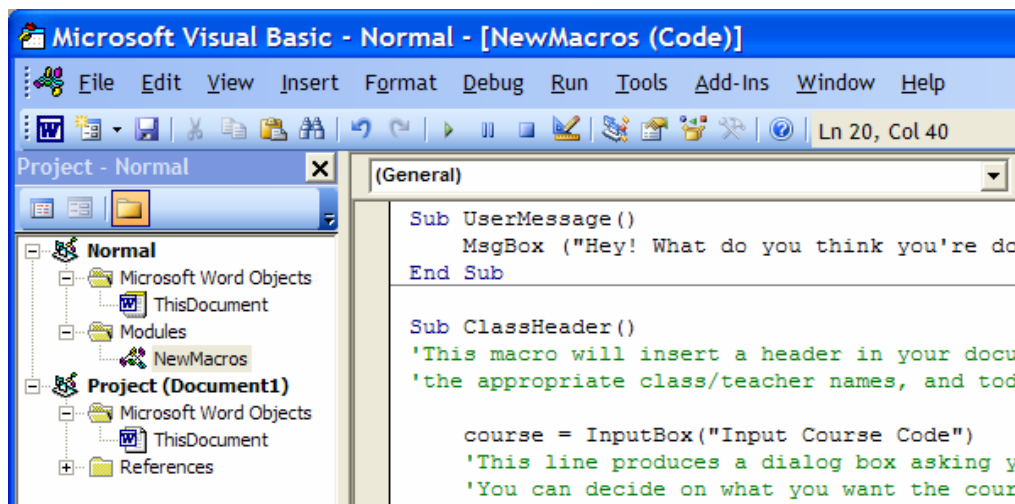
MACROS FOR THE LATIN CLASSROOM

(WORD 2003)

Visual Basic is the programming language in which macros are written. You may edit VB code as follows:

VISUAL BASIC EDITOR: Hit Alt-F11)

When the VB editor is opened, you should see the code for the macros stored in your Normal template. If you do not see any macro code, follow the steps below:



you
do

1. If there is a + sign next to "Normal" in the left hand frame, click on it.
2. Once "Normal" is expanded, you should see folders for MS Word Objects and Modules.
3. If there is a + sign next to "Modules" in the left hand frame, click on it.
4. Once "Modules" is expanded, you should see the "NewMacros" module. Double-click on it.

MACRO-WRITING GUIDELINES:

1. All macros must begin with "**Sub MacroName ()**" and must end with "**End Sub**". When you choose a macro name, remember that you may not use spaces or special characters.
2. Macro code governing font formatting will begin with "**Selection.Font**" and macro code governing paragraph formatting will begin with "**Selection.ParagraphFormat.**"
3. You will notice that as you type, you are given options for what might come next. Try typing "Selection.Font." or "Selection.ParagraphFormat." and scroll through the list of choices.
 - a. **You must type a period** after "Font" or "ParagraphFormat." This will cause the list of options to appear.
4. The sign is used to assign values to various parameters.
 - a. For example: **Selection.Font.Color = wdColorAqua**
 - b. Once you type the = sign, you will be given a list of options automatically.
5. "**With**" is used to set up a category of changes. If you open up the VB editor to work on a macro you've recorded, you will see code written as on the right rather than as on the left below. You do not need to use "With" when writing your own code, but you should know what it does so that you can understand the code Word generates.

Selection.Font.Name = "Times New Roman"
Selection.Font.Size = 12
Selection.Font.Bold = True
Selection.Font.Italic = False
Selection.Font.AllCaps = True

With Selection.Font
 .Name = "Times New Roman"
 .Size = 12
 .Bold = True
 .Italic = False
 .AllCaps = True
 End With

MACROS FOR THE LATIN CLASSROOM

(WORD 2003)

MESSAGE BOX MACRO:

- Message boxes can 1) display a message to the user or 2) ask yes/no questions and then take action on the answer.

INPUT BOX MACRO:

- Input boxes prompt the user to input some text, then the computer uses that information to take some action.

If you want to insert a **line break** in a message box to have the text run onto more than one line, enter **& Chr(13) &** between chunks of text. See the two examples below.



```
Sub MMA()  
  MsgBox ("Hello." & Chr(13) & "I am Mini Miss." & Chr(13) & "I am an action figure.")  
End Sub
```

```
Sub MMA()  
  MsgBox ("Hello." & Chr(13) & _  
    "I am Mini Miss." & Chr(13) & _  
    "I am an action figure.")  
End Sub
```

The character is used in VB to indicate that the command is continued on next line.

If you want to use text entered by the user in a message box or an input box, simply enter the name of the variable without quotes around it.

```
Sub MMA()  
  yourname = InputBox("Enter your name.")  
  MsgBox ("Hello." & Chr(13) & _  
    "I am Mini Miss." & Chr(13) & _  
    "I am an action figure." & Chr(13) & _  
    "And you are " & yourname & ".")  
End Sub
```



yourname is a variable. When the user enters something in the input box, this string of letters will become the value of *yourname*.

The `&` symbol indicates that more text is coming.

This is the name of the variable used to represent whatever the user enters in the input box.

MACROS FOR THE LATIN CLASSROOM

(WORD 2003)

You will find below some examples of MsgBox and InputBox applications:

```
Sub UserMessage()  
MsgBox ("Hey! What do you think you're doing?!")  
End Sub
```

```
Sub ClassHeader()  
'This macro will insert a header in your document with your name,  
'the appropriate class/teacher names, and today's date  
  
course = InputBox("Input Course Code")  
'This line produces a dialog box asking you to input the course code  
'You can decide on what you want the course codes to be, for example:  
'EN = English, LA = Language, MA = Math, SC = Science, HI = History  
  
If course = "LA" Or course = "la" Then  
'This section tells the computer what to do if LA or la is entered by the user  
  
    ActiveWindow.ActivePane.View.SeekView = wdSeekCurrentPageHeader 'View header  
  
    Selection.WholeStory 'Select everything that is already in the header  
    Selection.Delete 'Delete everything that is already in the header  
  
    Selection.ParagraphFormat.Alignment = wdAlignParagraphRight 'Right-align paragraph  
  
    Selection.Font.Size = 10 'Font size = 10  
  
    Selection.TypeText Text:="Maximus Decimus Meridius" 'Student Name  
    Selection.TypeParagraph 'Line break  
    Selection.TypeText Text:="Latin 1" 'Course Title  
    Selection.TypeParagraph 'Line break  
    Selection.TypeText Text:="Ms. Andresian" 'Teacher Name  
    Selection.TypeParagraph 'Line break  
End If  
  
NewMacros.CheckDate 'Runs the CheckDate macro below.  
  
ActiveWindow.ActivePane.View.SeekView = wdSeekMainDocument 'Close header  
  
End Sub
```

```
Sub CheckDate()  
'Gives the user the option to insert today's date or some other date.  
  
showdate = MsgBox("Do you want to enter today's date?", vbYesNo)  
'This line produces a message box asking the user a yes or no question.  
'The MsgBox will return a value of either 6 (YES) or 7 (NO)  
  
If showdate = 6 Then 'If the user has clicked the "YES" button Then...  
    Selection.InsertDateTime 'Today's date is inserted.  
  
ElseIf showdate = 7 Then 'If the user has clicked the "NO" button...  
    userdate = InputBox("Type the date as you wish it to appear.")  
                                'User is asked to enter a date.  
    Selection.TypeText Text:=userdate 'This date is then inserted.  
End If  
  
End Sub
```

You may add comments/explanations to your code by using the character `'` to indicate that what follows is a comment and not actual code.