COMP1022Q
Introduction to Computing with Excel VBA

Different Ways to Make VBA Macros

David Rossiter and Gibson Lam

Outcomes

- After completing this presentation, you are expected to be able to:

    1. Create a macro by recording one

    2. Explain the difference between a relative and absolute recorded macro

    3. Change the VBA code of a recorded macro

Creating Macros

- So far you have created a macro by *writing* a macro, i.e. writing your own VBA code

- Another way is by *recording* a macro

- In this way it's possible to make a macro without writing any code

- Recording a macro also gives us a useful way to learn some VBA code
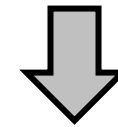
Creating Macros

- Sometimes you do this:

    - First, record the macro

    - then look at the VBA code to see
      what it is doing

    - then maybe adjust it or extend it so it behaves
      differently, does some more things, or does
   different things

Using Excel to Make a Macro

- You don't need to know how to do any VBA programming to make a macro

- That's because you can *record* a macro using the steps shown on the right

Start recording the macro

⬇

Everything you do is converted into a sequence of VBA code

⬇

Stop recording the macro

- When you finish recording you can then use the macro by running it, as many times as you like
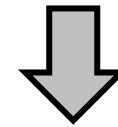
Two Types of Recorded Macro

- There are two types of recorded macro:
  - A macro which uses absolute references
    - For example, an absolute reference is B2
    - Because it uses absolute references, this type of macro will work only on one particular place in the worksheet
  - A macro which uses relative references
    - For example, a relative reference is B2
    - Because it uses relative references, this type of macro can be applied anywhere in the worksheet
- In this presentation we'll look at macros which use relative references

The Timetable Example

- We will use this timetable for our example:
- The cells look very boring and unattractive
- We can use macros to make them much nicer
- We will use a relative reference macro, so the macro can be applied anywhere

|	A	B	C	D	E	F
4			My Timetable			
5						
6		Monday	Tuesday	Wednesday	Thursday	Friday
7						
8	0900-0930	CHEM1202 Lab, 7141				
9	0930-1000		COMP2101, 3405		COMP2101, 3405	
10	1000-1030					
11	1030-1100					
12	1100-1130					
13	1130-1200	ECON1103, 4401		ECON1103, 4401		COMP2101 Lab, 4211
14	1200-1230					
15	1230-1300					
16	1300-1330	SOSC2011, 2408		SOSC2011, 2408		
17	1330-1400		LANG3102, 5507		LANG3102, 5507	
18	1400-1430					
19	1430-1500					
20	1500-1530			ECON1103 Tutorial, 3304		
21	1530-1600		CHEM1202, 2464		CHEM1202, 2464	
22	1600-1630					
23	1630-1700					
24	1700-1730					
25	1730-1800					
26	1800-1830					
27	1830-1700					

Changing the Formatting of a Course

- We first choose one of the courses, e.g. CHEM1202

- We want to make a macro which will help us apply exactly the same visual formatting to each of the CHEM1202 cells

- To get started, select one of the CHEM1202 cells

Creating a Macro for the Course

- We will now create a macro which will work in any cell (the cell which is currently selected)

- To do that we first make sure '*Use Relative References*' is selected
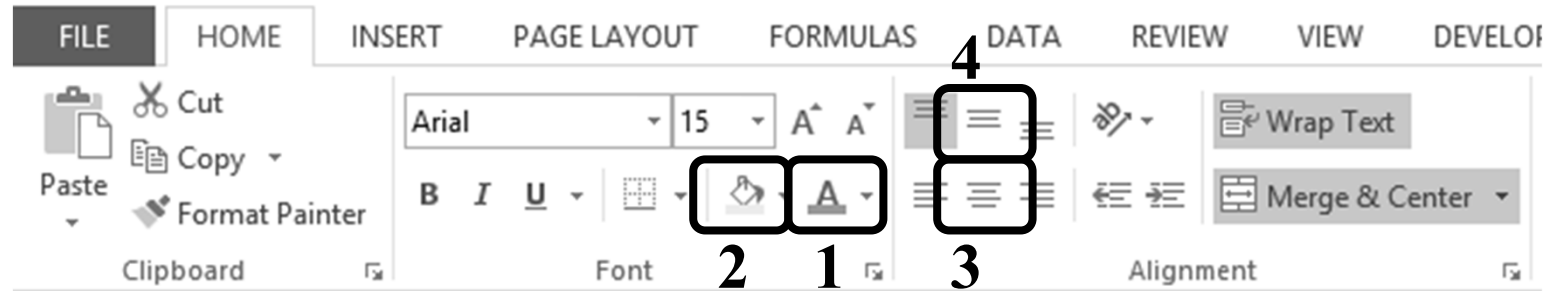
- Then, click on '*Record Macro*'
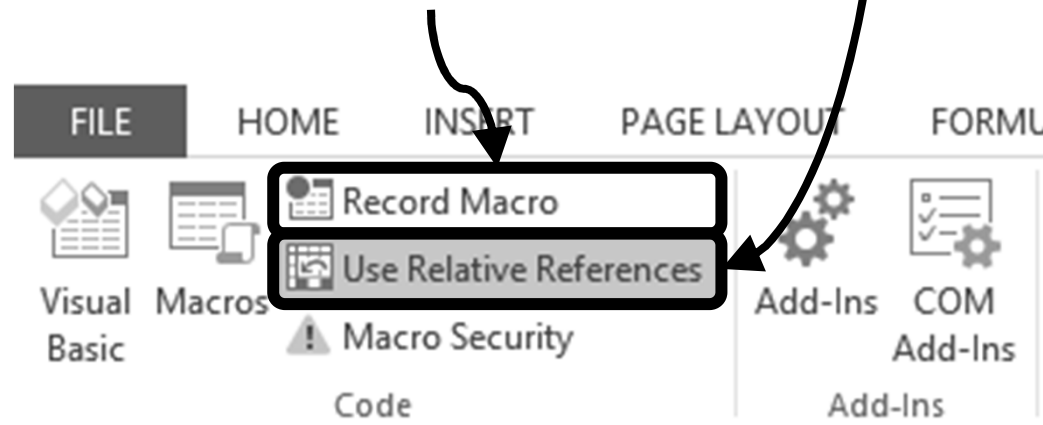
Preparing to Record the Macro

- The following window will appear

- Just like the way to you make your macros, you can change the name of the macro and the shortcut key you want to press to run the macro

- After you press OK, everything you click and do inside Excel is recorded
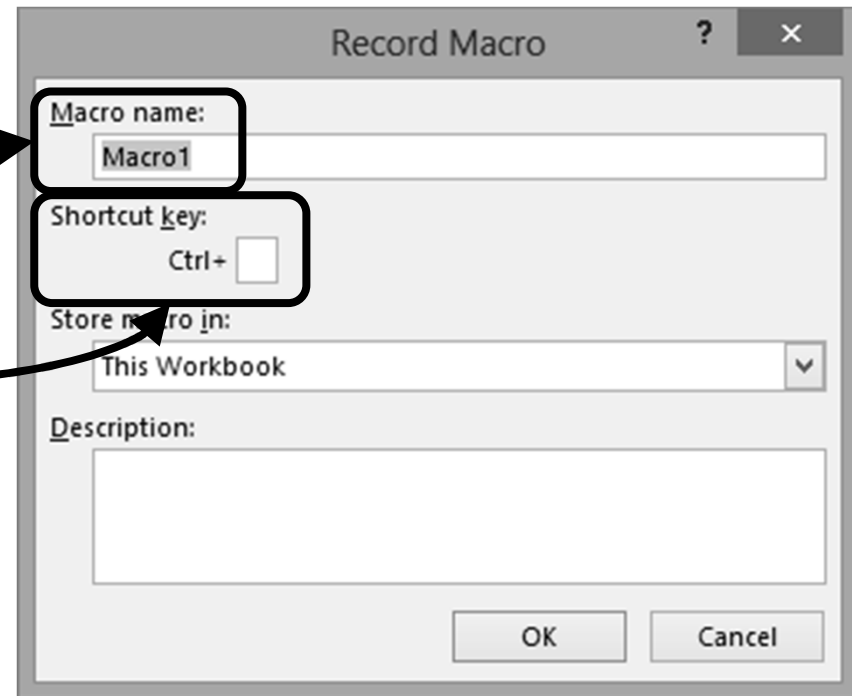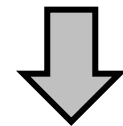
Actions Recorded in the Macro

- Now do whatever formatting you want to apply, for example:

  1. Set the text colour

  2. Set the background colour

  3. Center the text in the x axis

  4. Center the text in the y axis

- These are just some examples, you can do as many things to the cell as you like

Finishing the Recording
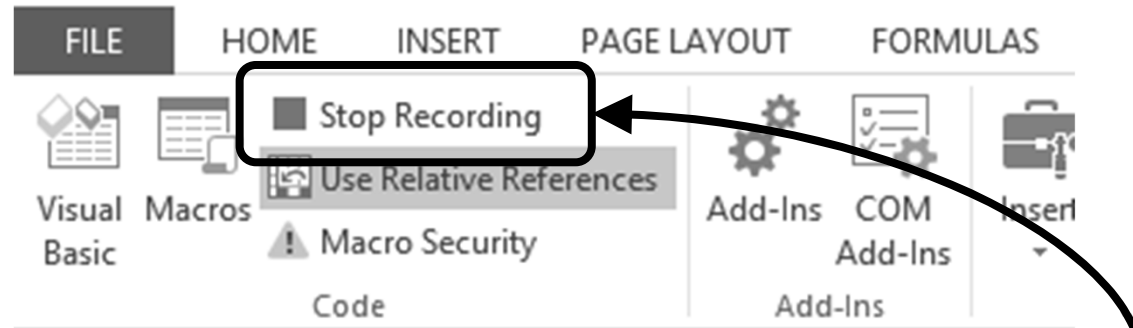
- After you have done all the formatting, stop the recording process by selecting '*Stop Recording*'

- You have finished making the macro

- In the next few slides, we will look at the VBA code of the newly created macro

Looking at the Code

- After you have made a macro, it would be a good idea to check it

- To do that, select the macro you have just created from the Macro window

- Click on *Edit* and the code of the macro is shown

You can also press Alt-F8
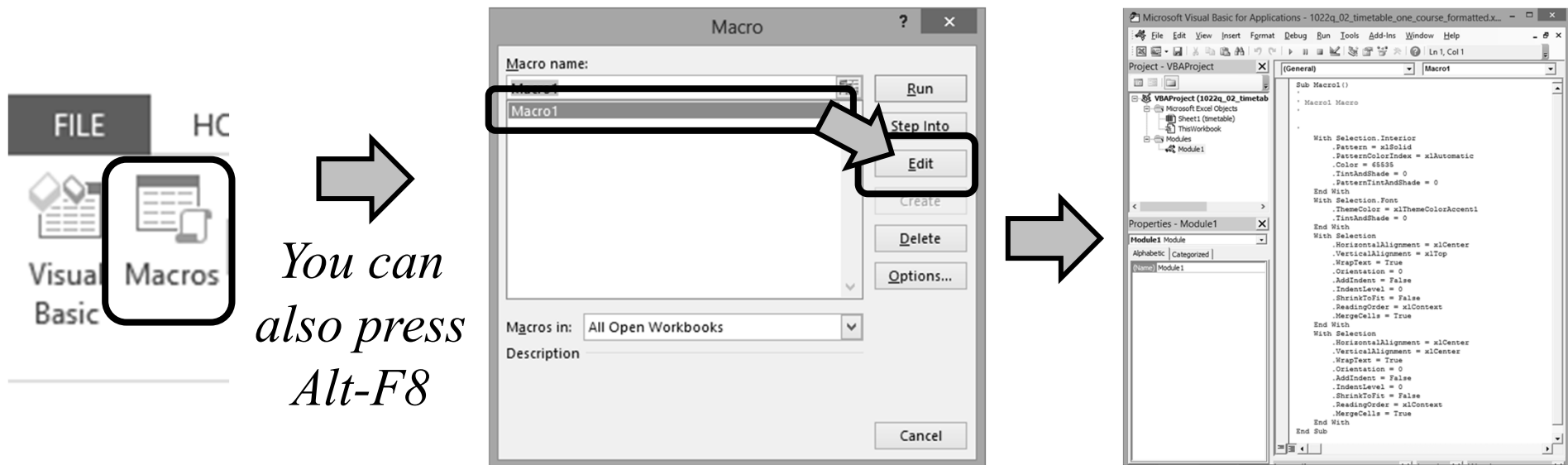
The Macro Code

- The code of the macro is shown in the VBA editor


```
Microsoft Visual Basic for Applications - 1022q_02_timetable_one_course_formatted.x...

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

Ln 1, Col 1

Project - VBAProject

VBAProject (1022q_02_timetab
  Microsoft Excel Objects
    Sheet1 (timetable)
    ThisWorkbook
  Modules
    Module1

Properties - Module1

Module1 Module

Alphabetic   Categorized

(Name)  Module1

(General)                          Macro1

Sub Macro1()
'
' Macro1 Macro
'

'

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 65535
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
    With Selection.Font
        .ThemeColor = xlThemeColorAccent1
        .TintAndShade = 0
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlTop
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
End Sub
```
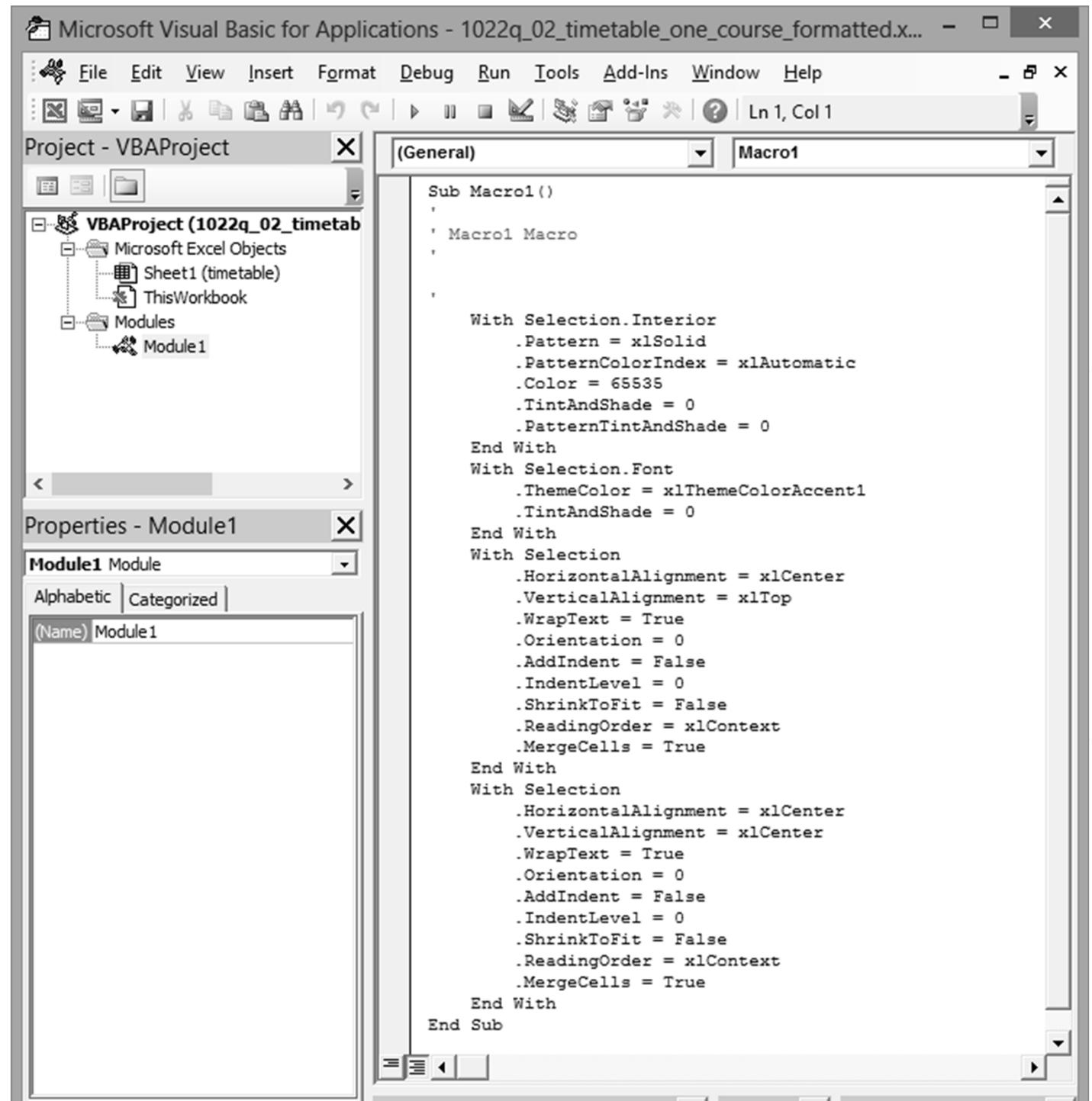
Comments in VBA Code

- As you know, anything after a `'` is a *comment*

- Comments are commonly used as explanations of code, they are ignored when the program runs

- At the top of the macro code some comments are generated by Excel which don't say anything useful!

- You can delete them or change them if you want

```
Sub Macro1()
'
' Macro1 Macro
'
'

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xl
        .Color = 65535
        .TintAndShade = 0
        .PatternTintAndShade =
```

Recorded Actions in the Macro

- In this example, when this macro was made four Excel buttons were clicked (changing the colour, etc)

- That's why you can see four pieces of code in this macro

You don't need to understand every line of this VBA code

```vba
Sub Macro1()
'
' Macro1 Macro
'

'
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 65535
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
    With Selection.Font
        .ThemeColor = xlThemeColorAccent1
        .TintAndShade = 0
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlTop
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
End Sub
```

Formatting a Course Using the Macro

- Here the timetable shows our newly created macro has been applied twice (in addition to the cell which was changed when the macro was recorded)

| J28 | | | ⨯ ✓ fx | | | |

	A	B	C	D	E	F
1			My Timetable			
2						
3		Monday	Tuesday	Wednesday	Thursday	Friday
4						
5	0900-0930					
6	0930-1000	CHEM1202 Lab, 7141	COMP2101, 3405		COMP2101, 3405	
7	1000-1030					
8	1030-1100					
9	1100-1130					
10	1130-1200	ECON1103, 4401		ECON1103, 4401		COMP2101 Lab, 4211
11	1200-1230					
12	1230-1300					
13	1300-1330	SOSC2011, 2408		SOSC2011, 2408		
14	1330-1400		LANG3102, 5507		LANG3102, 5507	
15	1400-1430					
16	1430-1500					
17	1500-1530			ECON1103 Tutorial, 3304		
18	1530-1600					
19	1600-1630		CHEM1202, 2464		CHEM1202, 2464	
20	1630-1700					
21	1700-1730					
22	1730-1800					
23	1800-1830					
24	1830-1700					

A Macro for Each Course

- In the same way, you can make 4 more macros to format the other courses, and then apply them to the appropriate cells

- The result can look something like this:

	Monday	Tuesday	Wednesday	Thursday	Friday
			My Timetable		
0900-0930	CHEM1202 Lab, 7141	COMP2101, 3405		COMP2101, 3405	
0930-1000					
1000-1030					
1030-1100					
1100-1130					
1130-1200	ECON1103, 4401		ECON1103, 4401		COMP2101 Lab, 4211
1200-1230					
1230-1300					
1300-1330	SOSC2011, 2408		SOSC2011, 2408		
1330-1400		LANG3102, 5507		LANG3102, 5507	
1400-1430					
1430-1500					
1500-1530			ECON1103 Tutorial, 3304		
1530-1600		CHEM1202, 2464		CHEM1202, 2464	
1600-1630					
1630-1700					
1700-1730					
1730-1800					
1800-1830					
1830-1700					

Making Four More Macros

- To make four further macros,
  one way is to use the macro recorder four more times

- Another more clever way is by using the VBA code:
  - Copy the macro code that already exists
  - Paste it (underneath the current code)
  - Change the name of the macro code you just pasted
  - Change some of the macro code you just pasted to make
    the macro apply a different colour, etc

- This is illustrated on the following slides

Making a New Macro by Copying a Current Macro

1. Select the macro code, copy it

2. Select an appropriate place for the new code by clicking under the macro code, paste the code

3. Change the name of the macro you have just pasted to something appropriate

4. Make adjustments to the code depending on what you want it to do

• You have now finished making a new macro!

Things to Play With

```
Sub Macro1_CHEM1202()

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 65535
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
    With Selection.Font
        .ThemeColor = xlThemeColorAccent1
        .TintAndShade = 0
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlTop
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
End Sub
```

- The text colour is set here
- Although you don't know exactly what the number means, try changing it to any other number in the range 0-65535
- The background colour is set here
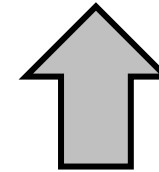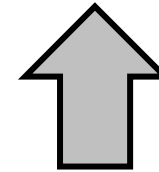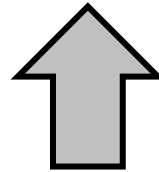- Try changing the '1' into a number in the range 1-6
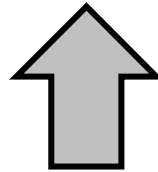
The Five Finished Macros


```
Sub Macro1_CHEM1202()

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 65535
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
    With Selection.Font
        .ThemeColor = xlThemeColorAccent1
        .TintAndShade = 0
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlTop
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
End Sub
```

```
Sub Macro2_COMP2101()

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 40404
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
    With Selection.Font
        .ThemeColor = xlThemeColorAccent4
        .TintAndShade = 0
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlTop
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
        .Font.Bold = True
    End With
End Sub
```
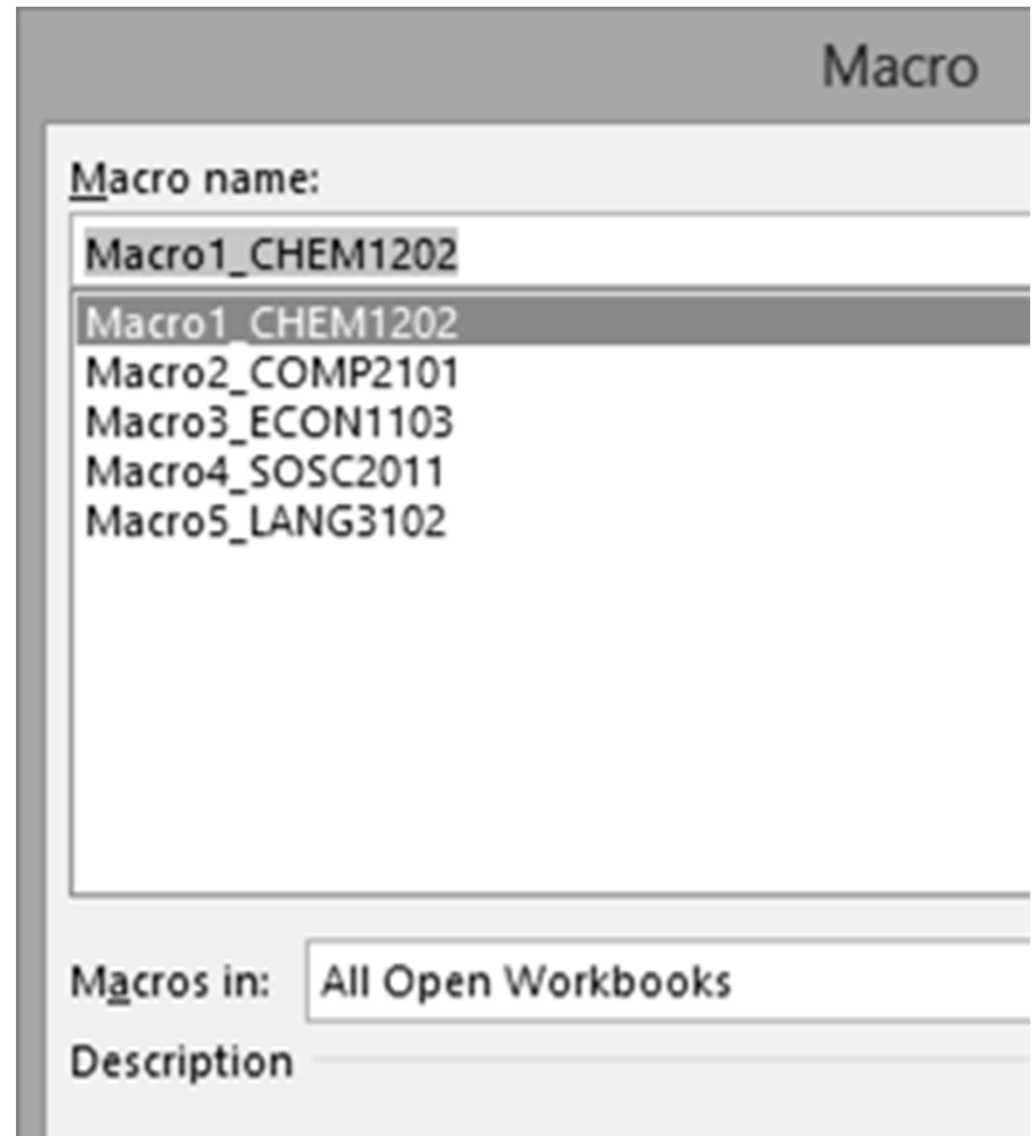
```
Sub Macro3_ECON1103()

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 1
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
    With Selection.Font
        .ThemeColor = xlThemeColorAccent1
        .TintAndShade = 0
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlTop
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
End Sub
```

```
Sub Macro4_SOSC2011()

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 60000
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
    With Selection.Font
        .ThemeColor = xlThemeColorAccent2
        .TintAndShade = 0
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlTop
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
End Sub
```

```
Sub Macro5_LANG3102()

    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .Color = 3728
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
    With Selection.Font
        .ThemeColor = xlThemeColorAccent6
        .TintAndShade = 0
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlTop
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With
End Sub
```

The Five Finished Macros

- Here you can see the five macros which have been created (one macro for each course)

- Each macro has been given an appropriate name, by editing the name shown in the VBA code

Macro

Macro name:

Macro1_CHEM1202

Macro1_CHEM1202
Macro2_COMP2101
Macro3_ECON1103
Macro4_SOSC2011
Macro5_LANG3102

Macros in: All Open Workbooks

Description

Starting a Macro

- Run a macro by selecting the macro, then selecting *Run*

- Or set up a keyboard shortcut by selecting the macro, selecting *Options*, then adding a Shortcut key, and use that

- Or click inside the subroutine and select the triangle
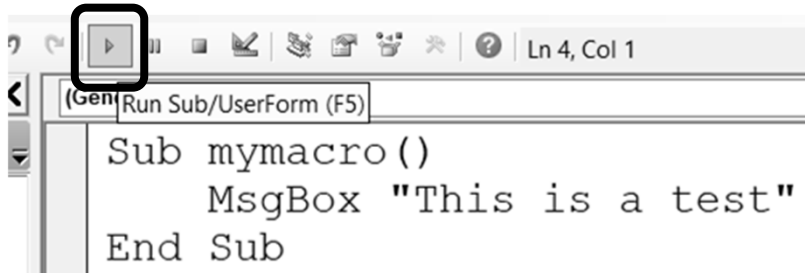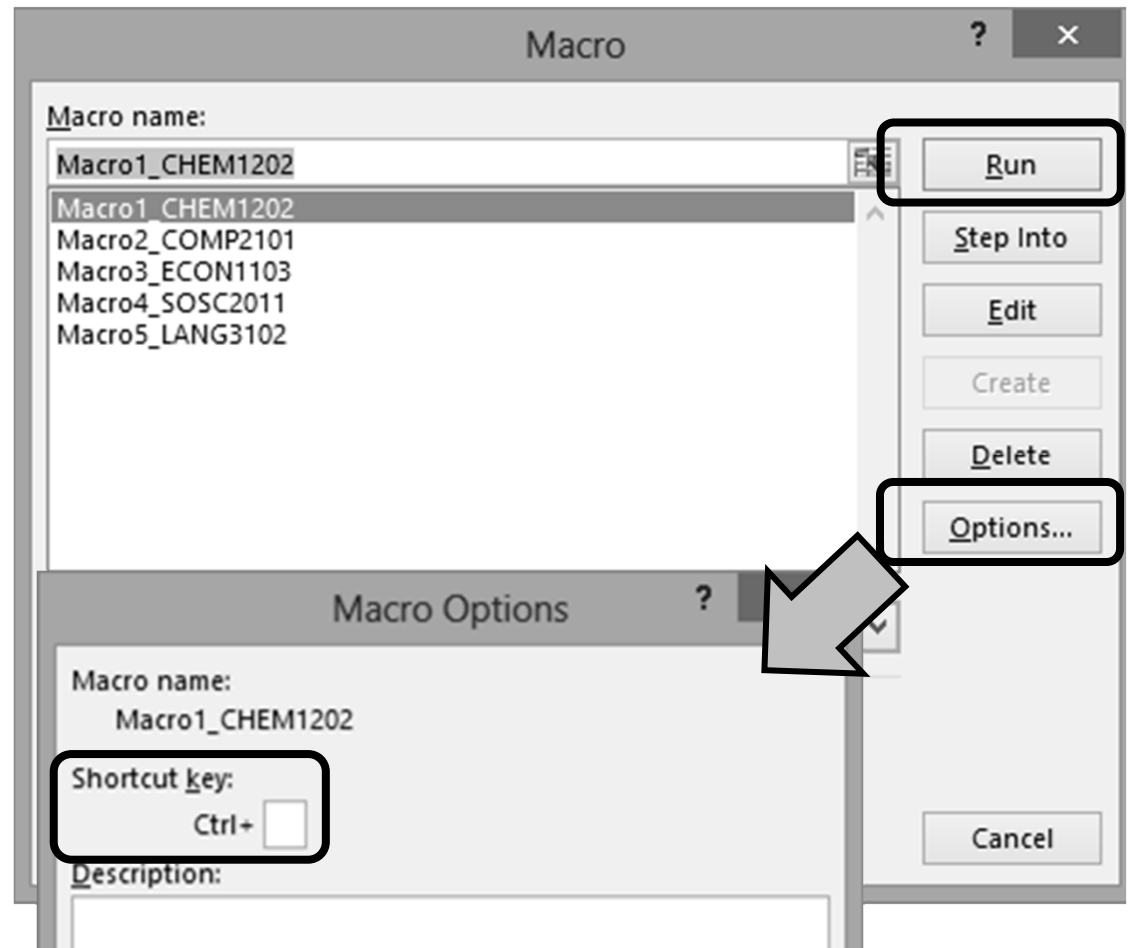
- Or use a shape to trigger the macro