COMP1022Q
Introduction to Computing with Excel VBA

Using Timer

Gibson Lam

Outcomes

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

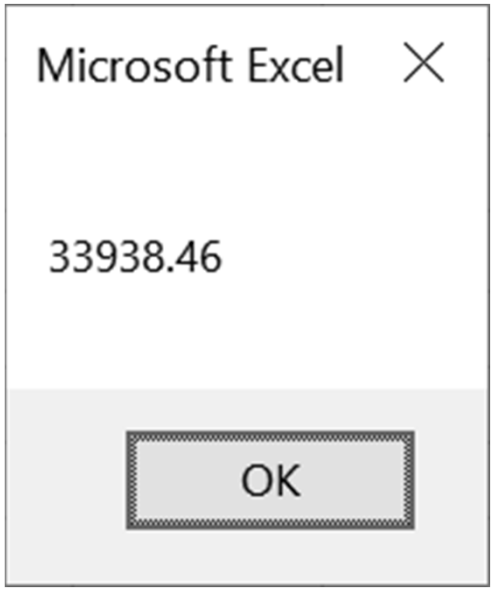
  1. Use the VBA Timer function
  2. Create countdown timers using VBA

Using Timer

- Timer is a VBA function that returns the time of the day as number of seconds since midnight
- Here is an example:

```
MsgBox Timer
```

- The value can be useful if you want to make programs that work with time

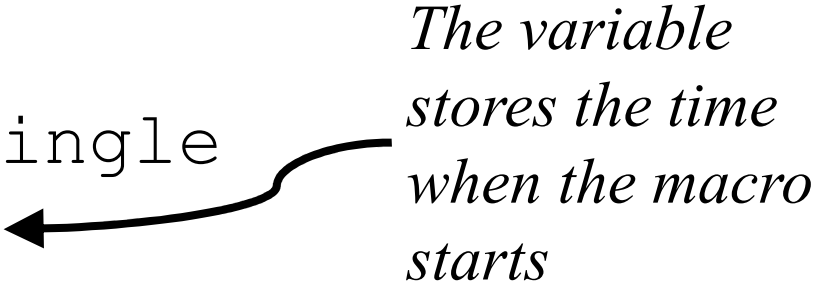


Showing the time value around 9:25am

Making a 3-second Timer

- You can easily make a countdown timer of three seconds in a macro using a combination of a loop and the `Timer` function
- First, you remember the time that you run the `Countdown` macro, like this:

```
Sub Countdown()
    Dim Start As Single
    Start = Timer
    ...
End Sub
```

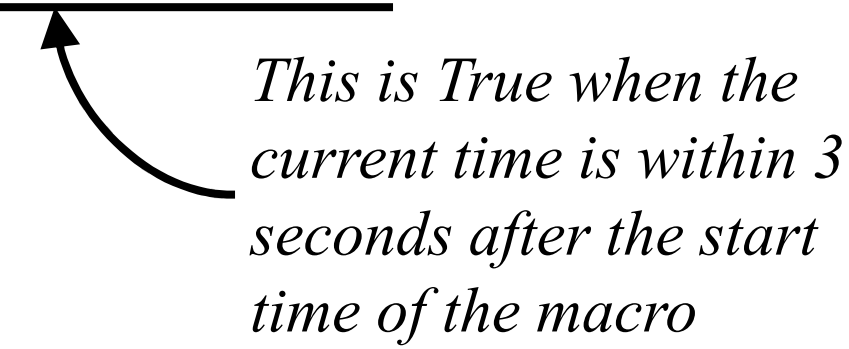


Using a Do While Loop

- A Do While loop can then keep on running until 3 seconds have passed, i.e.:

```
Do
    Now = Timer
Loop While Now < Start + 3
```

- After the loop finishes, you can then show a message, e.g.:

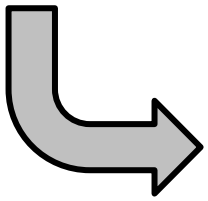



```
MsgBox "Wake up! Wake up! Wake up!"
```

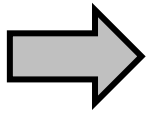
Running the Countdown Timer

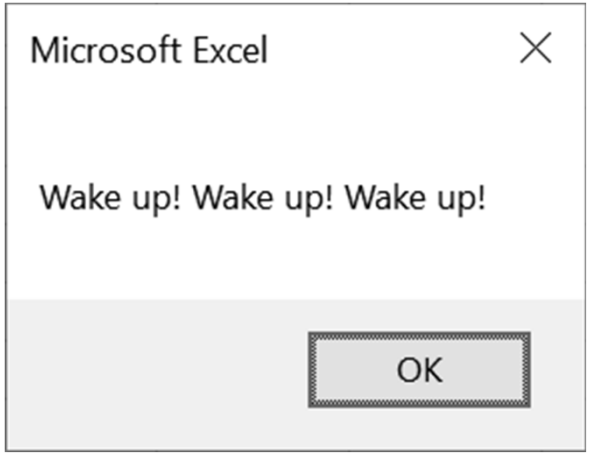
	A	B	C	D	E
1	**Example of a 3-second Countdown Timer**				
2	*This example makes a 3-second countdown timer*				
3					
4	*Press Ctrl-r to run the subroutine*				

Pressing Ctrl-r
to run the macro



3 seconds later…





Showing the Time

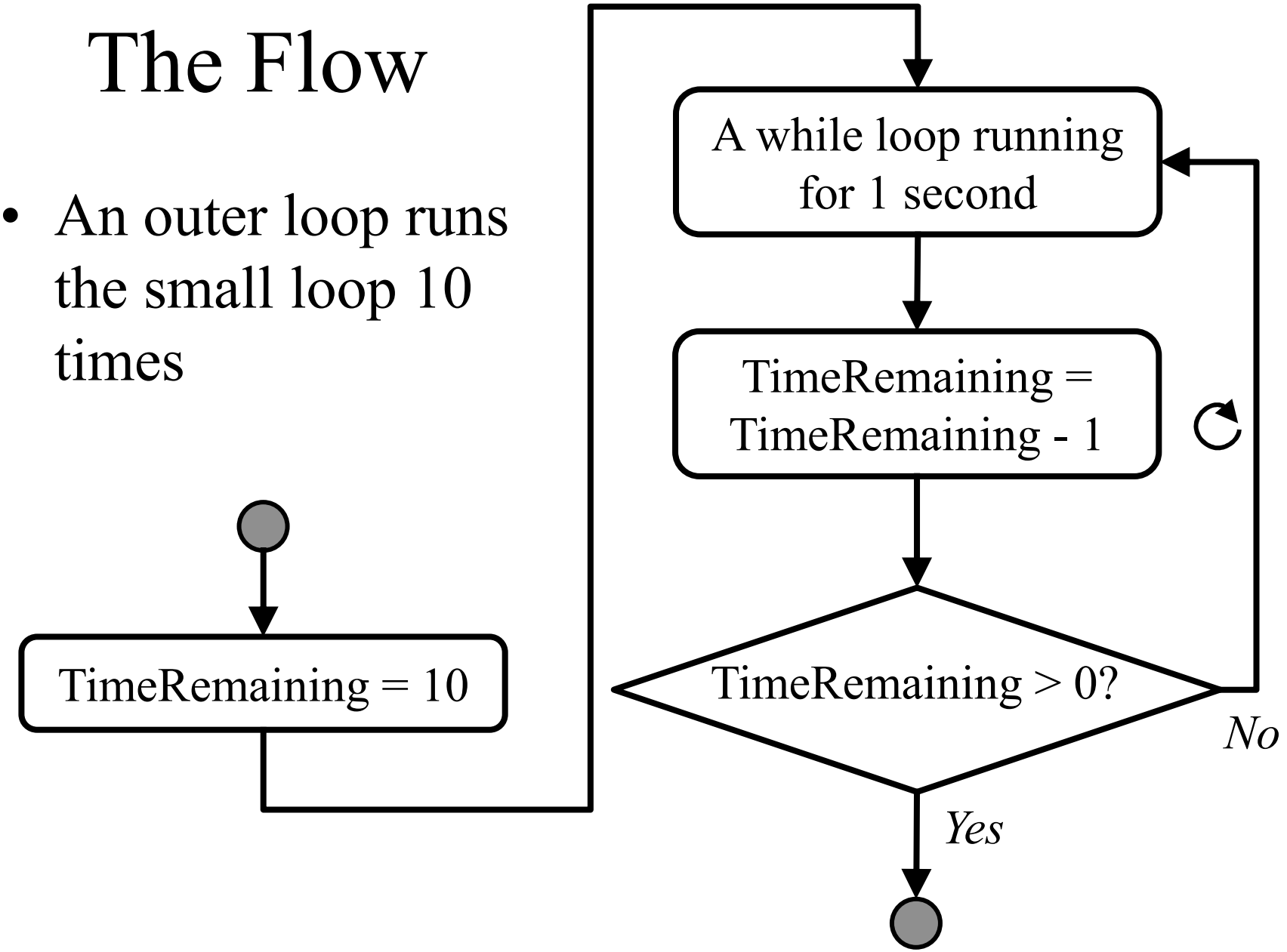
- It would be nice for the countdown timer to show the remaining time
- Let's make a 10-second timer with a remaining second display
- Rather than a 3-second while loop, we will use a 'shorter' 1-second loop, as shown below, and then run the loop 10 times, i.e.:

Run for 1 second ⟳

```
Do
      Now = Timer
Loop While Now < Start + 1
```

The Flow

- An outer loop runs the small loop 10 times



The Countdown Code

```
TimeRemaining = 10
Do
    Range("B4").Value = TimeRemaining

    Start = Timer
    Do
        Now = Timer
    Loop While Now < Start + 1

    TimeRemaining = TimeRemaining - 1
Loop While TimeRemaining > 0

Range("B4").Value = 0

MsgBox "Time's up!"
```

Run for 10 sec.

Show the remaining time in cell B4

Decrease the remaining time by 1

Using DoEvents

- If you run the countdown timer, you will find that the Excel program appears not responsive
- Sometimes, cell B4 may not even get updated correctly
- This is because VBA is too busy running the loop so that it has no time to do anything else
- To make sure Excel has time to do some other things, you can use the `DoEvents` command

Using DoEvents in the Countdown

- For example, you can use `DoEvents` inside the inner loop:

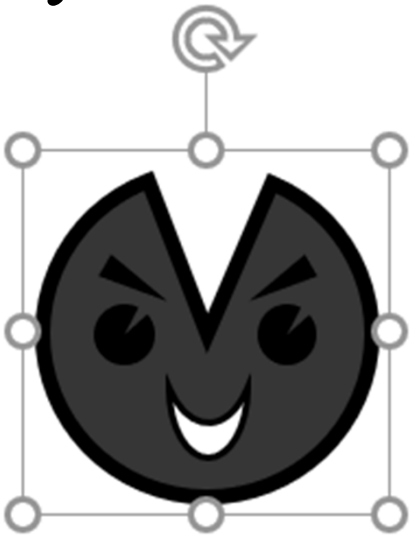
```
Do
      Now = Timer
      DoEvents
Loop While Now < Start + 1
```

With DoEvents, Excel can work on other things while VBA is busy running the loop

- The macro can now run and you can do other things with Excel at the same time

A Catching the Monster Game

- Using the countdown timer, a ‘Catching the Monster’ game can be created easily
- A monster is first drawn using simple Excel shapes
- When the shape is clicked, the following macro is run and a global variable `Caught` is set to True, i.e.:



```
Sub Catch()
    Caught = True
End Sub
```

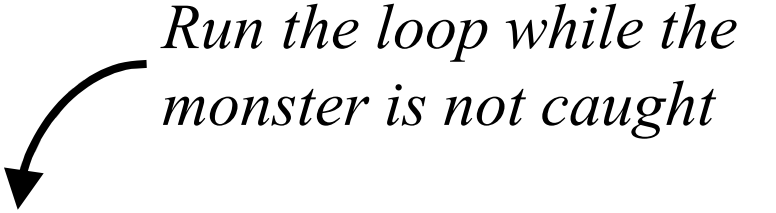
The variable shows the monster is caught or not

Modifying the Countdown Loop

- Then every time the countdown loop runs, the monster is moved to a new location:

```
Monster.Left = Rnd() * 530 + 50
Monster.Top = Rnd() * 85 + 75
```

- The main loop condition is changed to include the monster status, i.e.:




```
Do
    ...
Loop While Not Caught And _
          TimeRemaining > 0
```

Game Over

- Based on the value of `Caught`, you can show an appropriate game message after the loop finishes

```
If Caught Then
    MsgBox "Hooray! You caught the monster!"
Else
    Range("D14").Value = 0
    MsgBox "Time's up! Game over!"
End If
```