

COMP1022Q
Introduction to Computing with Excel VBA

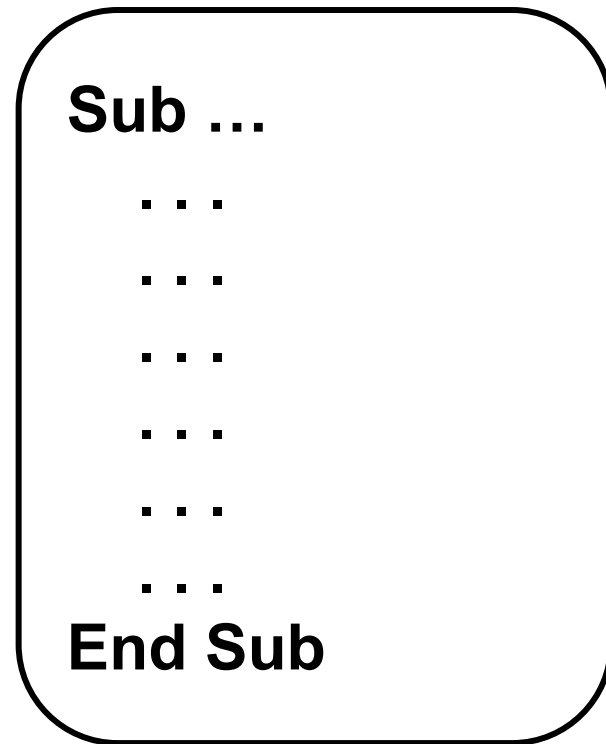
VBA Subroutines and Functions

David Rossiter and Gibson Lam

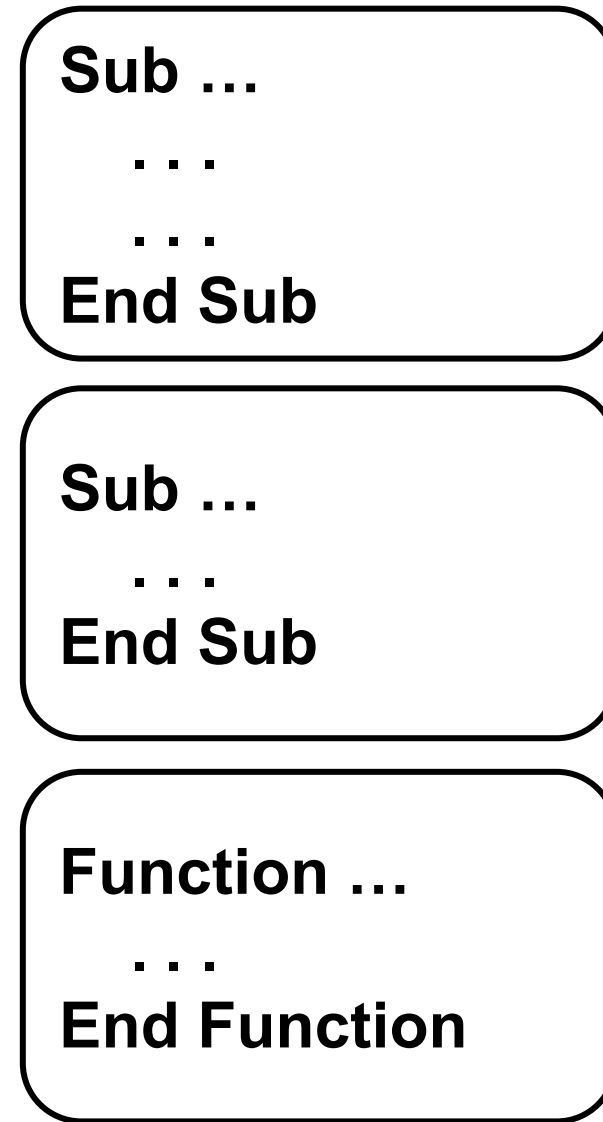
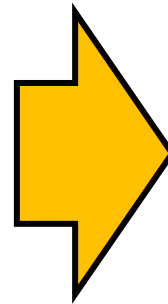
Outcomes

- After completing this presentation, you are expected to be able to:
 1. Group code into a VBA subroutine or function
 2. Explain the difference between a subroutine and a function
 3. Know how to finish a subroutine or a function early

Better Code Design



One big piece of code
is hard to manage

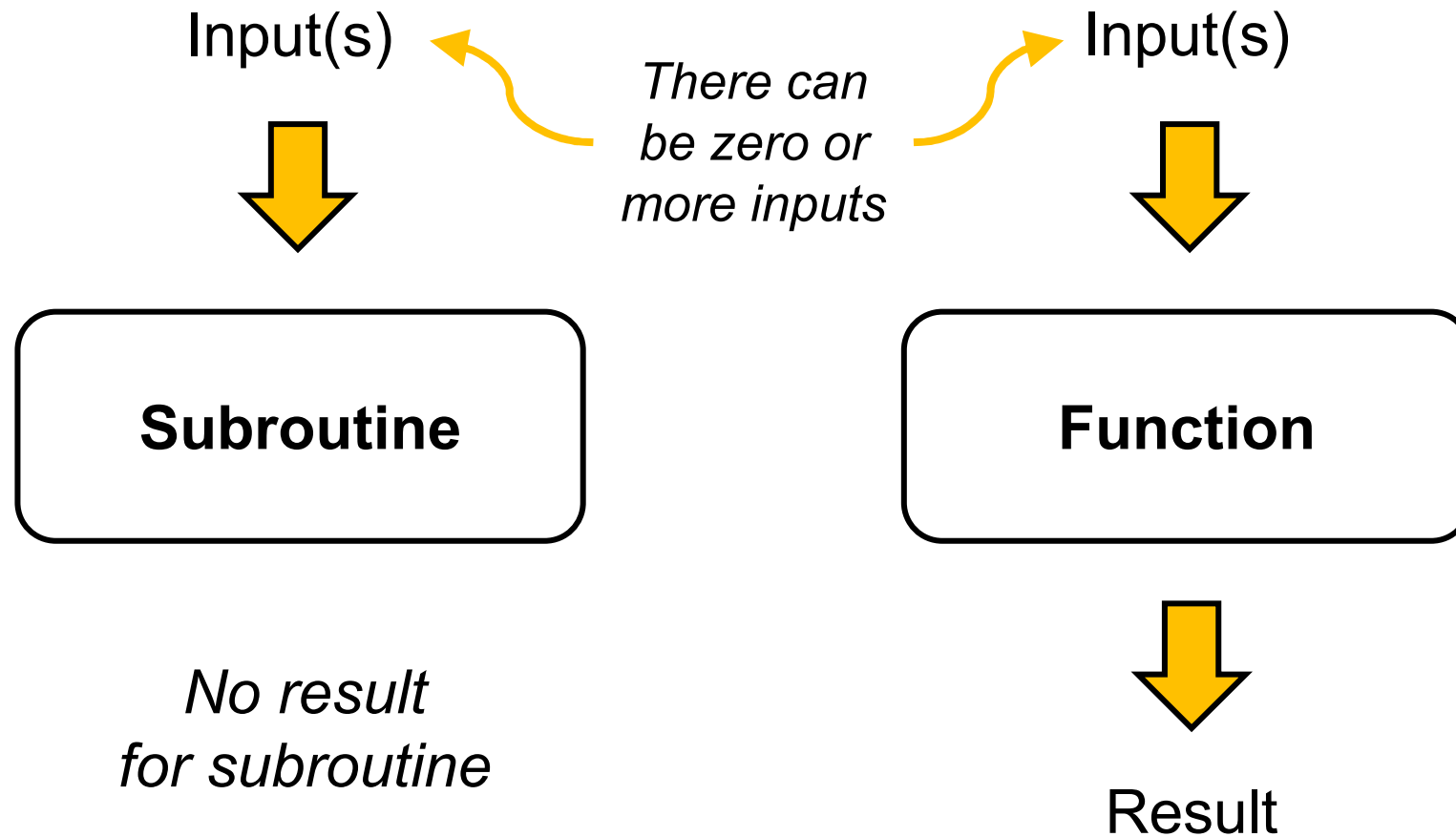


Usually we divide it into several
subroutines/ functions, for more
efficient handling

VBA Subroutines and Functions

- VBA *subroutines* and *functions* are used to group together VBA code so that the code can be used any number of times
 - For example, a macro is a subroutine that can easily be used lots of times
- A VBA *subroutine* runs some code and **does not return anything**
- A VBA *function* runs some code and **returns a result**

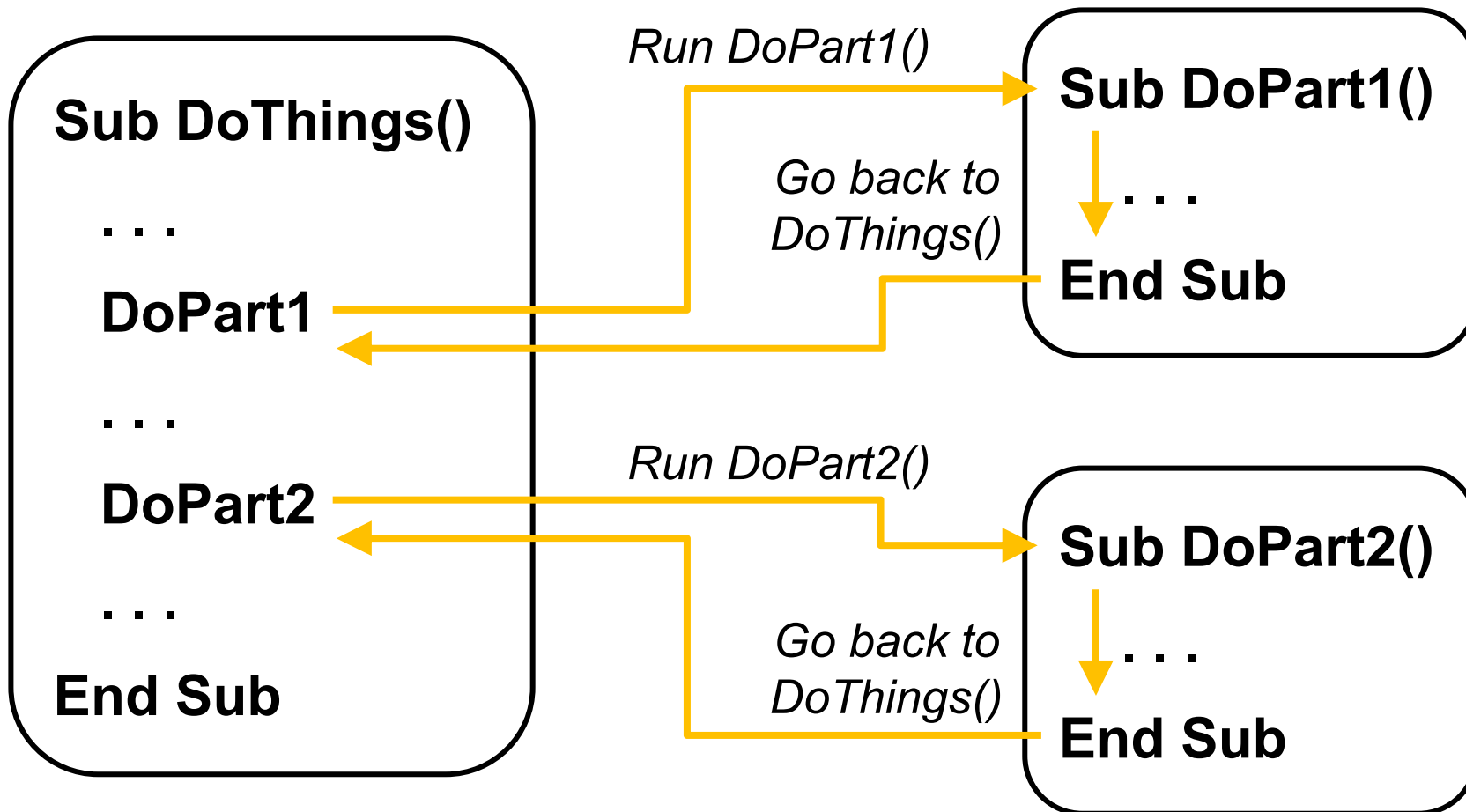
VBA Subroutines and Functions



Running a Subroutine

- You already know several methods to run a subroutine e.g. use a keyboard shortcut so that you can run a macro by pressing a key
- Another way to run a subroutine is from another subroutine
- An example is shown on the next slide to run two subroutines from a subroutine

Example of Running 2 Subroutines



VBA Subroutines

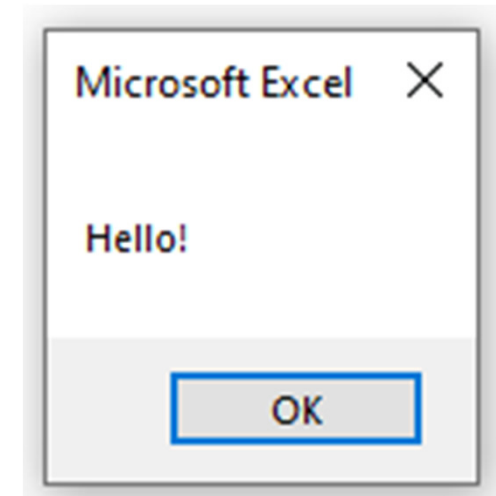
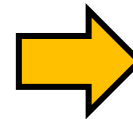
- At this point in the course you have used lots of subroutines – a macro is a subroutine
- An example: `Sub SayHello()`

`MsgBox "Hello!"`

`End Sub`

- You can run the subroutine by typing the name:

`SayHello`



Zero or more
Input(s)



Subroutine

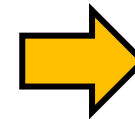
Passing a Value

- If we want to, we can pass something to the subroutine, like this:

```
Sub SayHello (Name)  
    MsgBox "Hello " & Name & "!"  
End Sub
```

- Using the subroutine:

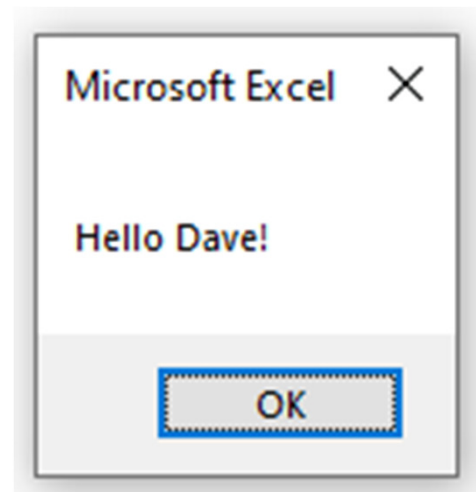
SayHello "Dave"



Zero or more
Input(s)



Subroutine



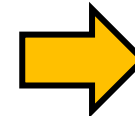
No Need to 'Dim'

- You don't have to use *Dim* to make the variable, and the variable name could be anything:

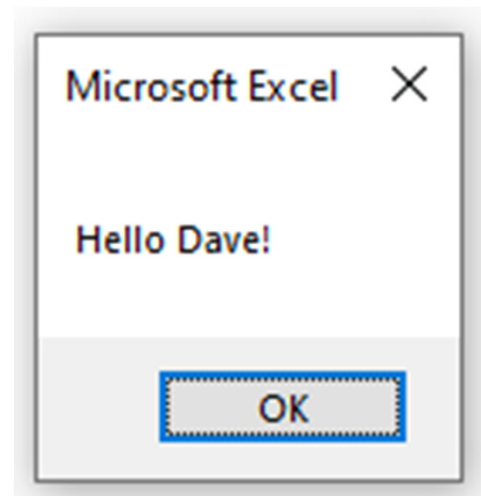
```
Sub SayHello(TextToShow)
    MsgBox "Hello " & TextToShow & "!"
End Sub
```

- Using the subroutine:

```
SayHello "Dave"
```



Zero or more
Input(s)



Passing More Values

- You can have as many inputs as you want
- In this example there are 2 inputs

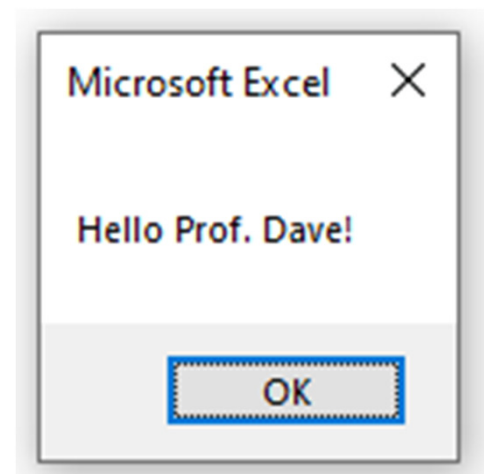
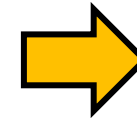
Zero or more
Input(s)



Subroutine

```
Sub SayHello(Title, Name)  
    MsgBox "Hello " & Title & " " & Name & "!"  
End Sub
```

SayHello "Prof.", "Dave"



Finishing a Subroutine Early

- Sometimes you want to finish a subroutine early
- You can use `Exit Sub` in the middle of the subroutine, e.g.:

```
Sub SubName ( ... )
```

```
    ...
```

```
    Exit Sub
```

```
    ...
```

```
End Sub
```

*The subroutine immediately finishes
when it sees this line of code.*

*The rest of the code in the
subroutine will not be executed.*

Example of Finishing a Sub Early

- This function uses `Exit Sub` to ‘give up’ when a number outside the range 1 to 100 is given to it

```
Sub HandleGuess()
```

```
    Dim ThisInput As Integer
```

```
    ThisInput = InputBox("Your lab 3 guess?")
```

```
    If ThisInput < 1 Or ThisInput > 100 Then
```

```
        MsgBox "You are being annoying!"
```

```
        Exit Sub
```

```
    End If
```

```
    . . .
```

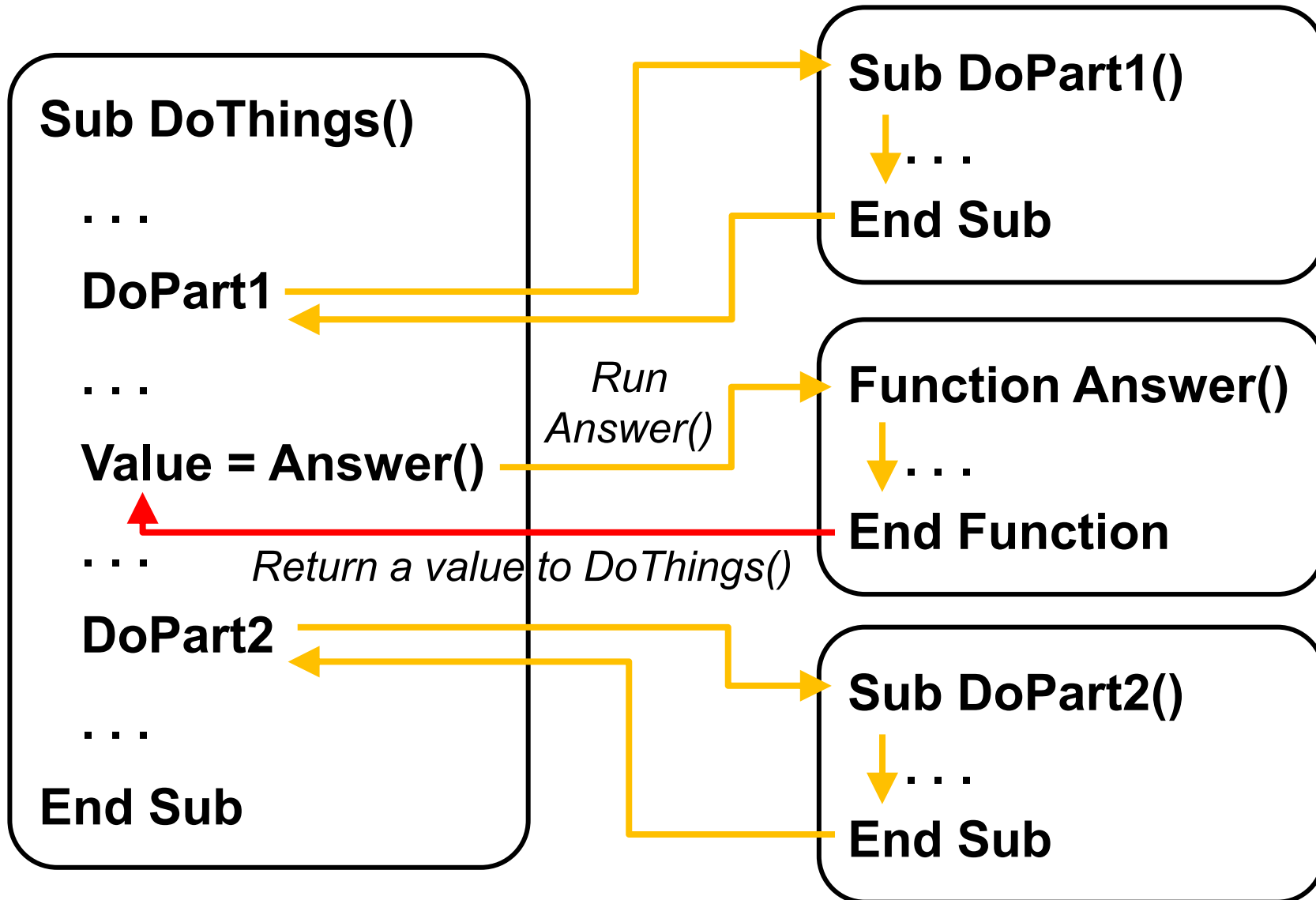
```
End Sub
```

*This subroutine finishes early when the
input is not in the range 1 to 100*

What About Functions?

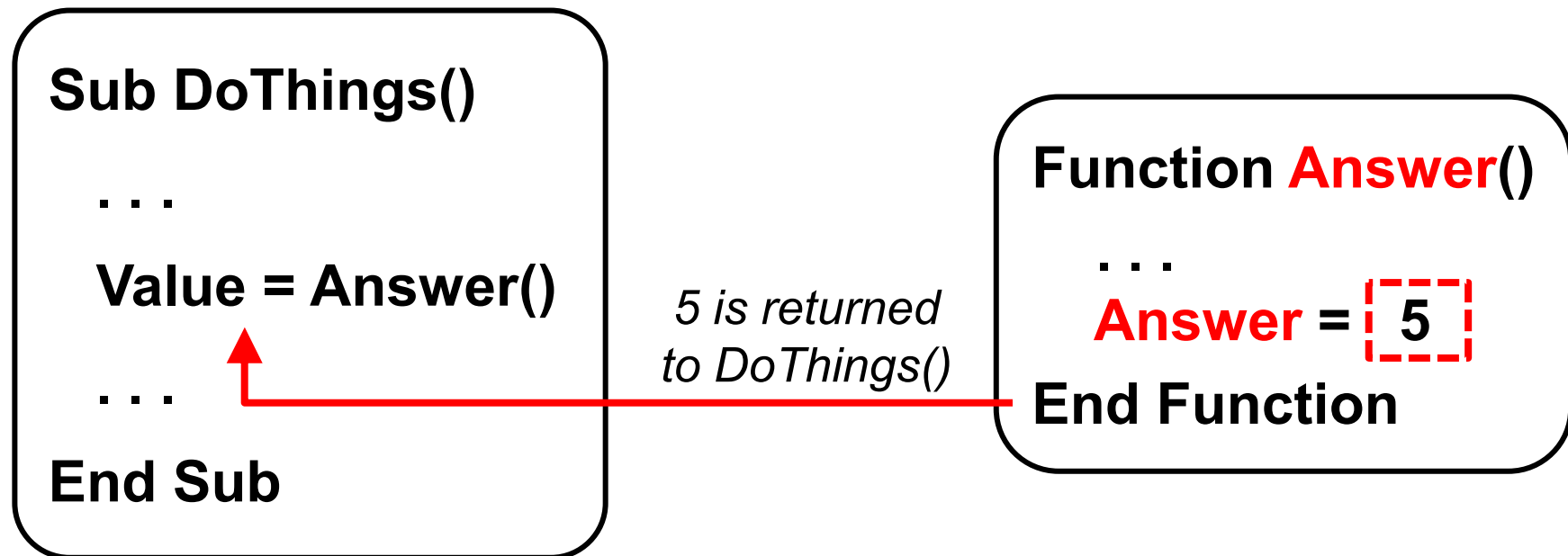
- A VBA Function is very similar to a VBA Subroutine
- A big difference is that a VBA Function returns an answer, where a VBA Subroutine doesn't
- An example is shown in the next slide using a function inside a subroutine

Example of Running a Function



A Special Variable

- A 'special' variable is given inside a function, which has the same name as the function
- This variable is used to store the value that gets returned by the function



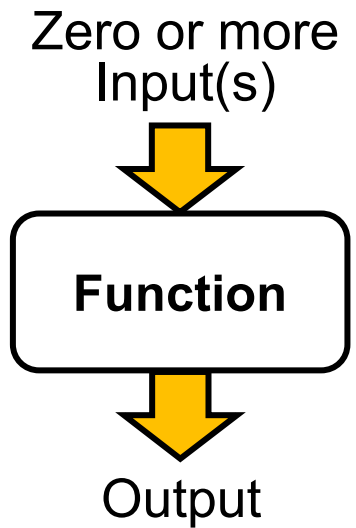
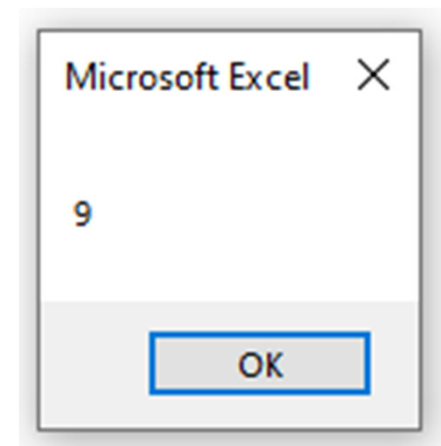
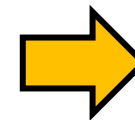
VBA Functions

- Here is a function which returns a random month number (1-12):

```
Function ChooseRandomMonth()  
    ChooseRandomMonth =  
        Int(Rnd * 12) + 1  
End Function
```

- Example of using the function:

```
Result = ChooseRandomMonth  
MsgBox Result
```



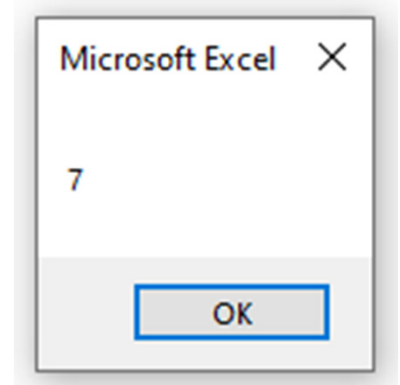
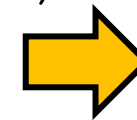
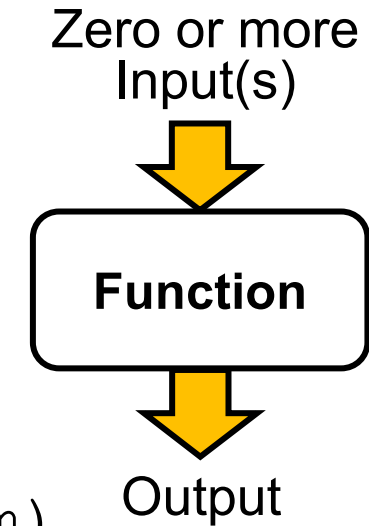
Passing a Value

- If we want to, we can pass something to the function, like this:

```
Function MakeRandomInteger (MaxNum)  
    MakeRandomInteger =  
        Int (Rnd * MaxNum) + 1  
End Function
```

- Example of using the function:

```
Result = MakeRandomInteger (12)  
MsgBox Result
```



Passing More Values

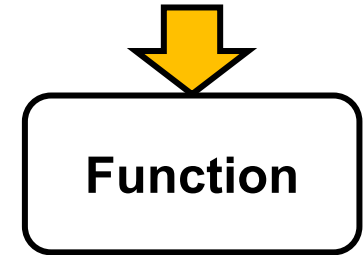
- You can have as many inputs as you want
- In this example there are 2 inputs

```
Function BuildName(Surname, OtherNames)  
    BuildName = Surname & ", " & OtherNames  
End Function
```

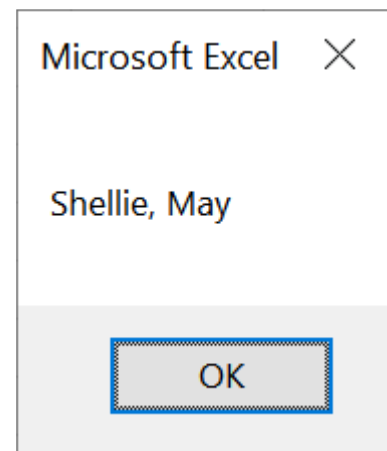
- Using the function:

```
Result = BuildName("Shellie", "May")  
MsgBox Result
```

Zero or more
Input(s)



Output

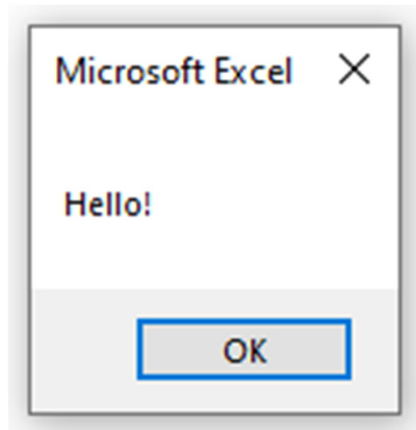


Functions Already in VBA

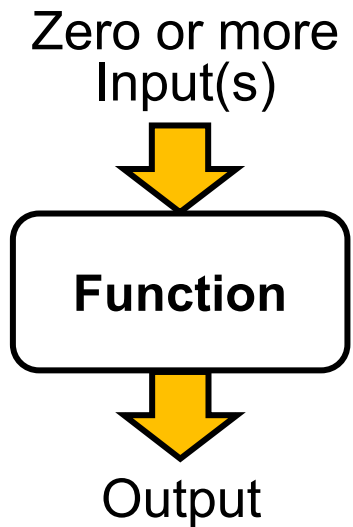
- There are functions already in VBA
- For example, InputBox, Left and Int are all functions
- An example of Left :

```
Result = Left("Hello! I am Dave!", 6)
```

MsgBox Result



Two input parameters are passed to the function



Finishing a Function Early

- Sometimes you want to finish a function early
- You can use `Exit Function` in the middle of the function, e.g.:

```
Function FuncName ( ... )
```

```
    ...
```

```
    Exit Function
```

```
    ...
```

```
End Function
```

The function immediately finishes when it sees this line of code.

The rest of the code in the function will not be executed.

Example of Finishing a Function Early

- This function calculates the square of a number
- It uses `Exit Function` to ‘give up’ when a number smaller than or equal to 0 is given to it

```
Function SquareSize(SideLength)
```

```
    If SideLength <= 0 Then
```

```
        SquareSize = 0
```

```
        Exit Function
```

```
    End If
```

*This function
finishes early
when the input
value is not a
positive number*

```
    SquareSize = SideLength * SideLength
```

```
End Function
```