



Macro Programming

Vignesh V

Department of Computer Applications

vigneshv@pes.edu

Macro Programming

User Defined Functions (UDFs) & Return Values

Vignesh V

Department of Computer Applications



User Defined Function (UDF)

- A **UDF** is a custom function written in VBA.
- It can be used in Excel formulas like `=SUM()` or `=AVERAGE()`.
- UDFs always **return a value**, unlike macros which perform actions.



Macro Programming

User Defined Function (UDF)

Function SquareNumber(x As Double) As Double

SquareNumber = x * x

End Function

Usage in Excel:

=SquareNumber(A1)



Difference Between Macros and UDFs

Aspect	Macros (Sub Procedures)	UDFs (Function Procedures)
Purpose	Automate actions (formatting, copying, etc.)	Perform calculations and return a result
Called From	Buttons, menus, or VBA code	Worksheet formulas or VBA
Return Value	None (Sub doesn't return a value)	Must return a value
Example	<code>Sub FormatReport()</code>	<code>Function GetDiscount()</code>



Function AddTwoNumbers(a As Double, b As Double) As Double

AddTwoNumbers = a + b

End Function



Creating a Simple UDF

Steps:

1. Open **VBE (Alt + F11)**.
2. Insert → **Module**.
3. Type the code above.

Go back to Excel and type:

`=AddTwoNumbers(10, 20)`



Creating a Simple UDF

Explanation:

Functions can take multiple parameters and return one value.



You can use ranges as inputs:

```
Function RangeSum(rng As Range) As Double
```

```
    RangeSum = Application.WorksheetFunction.Sum(rng)
```

```
End Function
```

- The argument type **Range** allows users to pass entire cell ranges from worksheets.



Returning Arrays from UDFs

You can return arrays, but must **enter the formula as an array formula** in Excel.

```
Function MultiplesOf(n As Integer) As Variant  
    Dim arr(1 To 5) As Integer, i As Integer  
    For i = 1 To 5  
        arr(i) = n * i  
    Next i  
    MultiplesOf = Application.Transpose(arr)  
End Function
```



Returning Arrays from UDFs

Steps:

1. Enter `=MultiplesOf(3)` in Excel.
2. Press **Ctrl+Shift+Enter** to return `{3;6;9;12;15}`.



Handling Errors in UDFs

Function SafeDivide(num As Double, denom As Double) As Variant

On Error GoTo ErrHandler

If denom = 0 Then

SafeDivide = "Error: Divide by zero"

Else

SafeDivide = num / denom

End If

Exit Function

ErrHandler:

SafeDivide = CVErr(xlErrValue)

End Function



Limitations of UDFs

UDFs **cannot**:

- Change cell formatting.
- Insert/delete rows or columns.
- Display message boxes.
- Modify other cells.

They are **purely for returning values** to the calling cell.



Experiential Task : Calculator

Task:

1. Create a UDF `Bonus(sales, target, rate)`
2. If `sales >= target`, return `sales * rate`; else return 0.

Example:

`=Bonus(B2, 1000, 0.1)`



Macro Programming

Vignesh V

Department of Computer Applications

vigneshv@pes.edu