



Macro Programming

Vignesh V

Department of Computer Applications

vigneshv@pes.edu

Macro Programming

Experiential Learning — Debugging VBA Code + Using Copilot to Automate Excel Tasks

Vignesh V

Department of Computer Applications



Types of VBA Errors

Error Type	Description	Example
Syntax Error	Breaks coding rules	<code>If x = 10 Then MsgBox "Hi"</code> (missing End If)
Runtime Error	Occurs while code runs	Dividing by zero, missing sheet
Logic Error	Code runs but gives wrong results	Incorrect formula or condition



Macro Programming

Types of VBA Errors

The screenshot shows the Microsoft Visual Basic for Applications editor window. The menu bar includes File, Edit, View, Insert, Format, Debug, Run, Tools, Add-Ins, Window, and Help. The toolbar contains various icons, including a green play button (Run) which is highlighted with a black box. The status bar at the bottom right indicates 'Ln 6, Col 8'. The (General) tab is selected, and the 'Run Sub/UserForm (F5)' button is highlighted in yellow. The code editor contains the following VBA code:

```
Sub SYNTAX_ERROR()  
MsgBox this is my first programme  
End Sub  
Sub myfirstprogram()  
MsgBox "this is my first programme"  
End Sub
```

A dialog box titled 'Microsoft Visual Basic for Applications' is displayed in the foreground. It features a yellow warning triangle icon and the text 'Compile error: Syntax error'. The 'Syntax error' text is highlighted in yellow. At the bottom of the dialog box are 'OK' and 'Help' buttons.



Experiential Task: Finding Syntax Errors

```
Sub SalesTotal()  
Dim total as Double  
For i = 1 To 10  
total = total + Cells(i, 2).Value  
MsgBox "Total Sales: " & total  
End Sub
```

Expected Fix: Add **Next i**



Macro Programming

Runtime Errors in VBA

Common runtime errors:

- Division by zero.
- Referring to an invalid worksheet or cell.
- Opening a missing file.



Macro Programming

Runtime Errors in VBA

Example:

```
Sub ExampleRuntime()  
    Dim result As Double  
    result = 100 / 0  
End Sub
```



Experiential Task: Handling Runtime Errors

Add structured error handling:

```
On Error GoTo ErrHandler
```

```
Dim result As Double
```

```
result = 100 / 0
```

```
Exit Sub
```

```
ErrHandler:
```

```
    MsgBox "An error occurred: " & Err.Description
```




Debugging Logic Errors

Logic errors are the hardest to find because VBA runs successfully but results are wrong.

```
Sub LogicErrorExample()  
    Dim x As Integer, y As Integer  
    x = 10  
    y = 0  
    Do Until x = 0  
        y = y + x  
        x = x - 2  
    Loop  
    MsgBox "Result: " & y  
End Sub
```



Debug systematically:

1. Reproduce the error.
2. Add breakpoints (F9).
3. Step through (F8).
4. Inspect variables (Immediate / Watch Window).
5. Fix and re-test.



Using Immediate Window

Use `?` to test variable values during runtime.

```
? x  
? Cells(1, 1).Value
```

Use `Debug.Print` to log information.

```
Debug.Print "Loop count: " & i
```



Steps:

1. Right-click variable → “Add Watch.”
2. Choose “Watch Expression.”
3. Observe value changes in Watch Window.



Breakpoints & Step Execution

Press **F9** to set breakpoint.

Run code → execution pauses.

Press **F8** to go line by line.

Observe variable changes in Locals Window.



Breakpoints & Step Execution

(General)

```
Sub BreakPoint()  
  
    Dim k As Long  
  
    Worksheets("Example1").Select  
  
    Range("A1").Select  
  
    Range("A1:A10").Value = "Hi threre!"  
  
End Sub
```



Introducing Microsoft Copilot



Copilot

Your everyday AI companion



Introducing Microsoft Copilot

Copilot is Microsoft's AI assistant that can help you:

- Generate VBA code snippets.
- Explain existing VBA logic.
- Suggest fixes for syntax or logic errors.
- Automate repetitive Excel tasks quickly.



Example: Using Copilot for VBA Help

Prompt Example:

“Write a VBA macro to highlight all cells with values greater than 500 in yellow.”



Macro Programming

Vignesh V

Department of Computer Applications

vigneshv@pes.edu