Experiential Learning Guide

Topic: Debugging VBA Code

Course: Macro Programming (UQ25CA142A)

Task:

1. Identify **syntax, runtime, and logic errors** in VBA programs.
2. Use **breakpoints**, **Step Into (F8)**, and **Step Over (Shift+F8)** to trace code execution.
3. Observe variable values in the **Immediate**, **Locals**, and **Watch** windows.
4. Apply **error-handling** using `On Error GoTo`.

What is Debugging?

Debugging means **finding and fixing mistakes (bugs)** in your program.
In VBA, you'll often encounter three types of errors:

Error Type	When It Happens	Example
Syntax Error	You break a rule of VBA grammar	Missing `Next` in a loop
Runtime Error	The code runs but fails during execution	Dividing by zero
Logic Error	The code runs, but gives a wrong result	Wrong formula or variable

Step 1: Experience — Run a Buggy Program

Write this code into a **new module** in your VBA editor:

```
Sub TotalSales()
    Dim i As Integer
    Dim total As Double
    For i = 1 To 10
        total = total + Cells(i, "B").Value
    Next i
    MsgBox "Total Sales is " & Total
End Sub
```

1. Press **F5** to run it.
2. Observe what happens — does it show an error or a wrong result?

Answer the following:

1. What kind of error is it (Syntax / Runtime / Logic)?
2. Which line caused the problem?
3. What message did Excel show?
4. What do you think might fix it?

Breakpoints

- Click the left margin of the code editor to set a **red dot (breakpoint)**.
- Run the macro again — it stops at that line.
- Press **F8** to go *one line at a time* (Step Into).

Immediate Window

- Open it via `View → Immediate Window` or press **Ctrl+G**.

You can type:

```
? total
```

- to display the current value of a variable.

Watch Window

- Right-click a variable → “Add Watch” → observe how its value changes when you step through.

Error Handling Example

```
Sub SafeDivide()
    On Error GoTo ErrHandler
    Dim x As Double, y As Double
    x = 10
    y = 0
    MsgBox x / y
    Exit Sub
ErrHandler:
    MsgBox "Error: " & Err.Description, vbCritical
End Sub
```

Run this and see how VBA handles the error.

Code 1 – Syntax Error

```
Sub MissingNext()
    Dim i As Integer
    For i = 1 To 5
        MsgBox i
    ' Missing Next i
End Sub
```

Task: Fix the syntax error and run again.

Code 2 – Runtime Error

```
Sub DivideByZero()

    Dim a As Integer, b As Integer

    a = 10

    b = 0

    MsgBox a / b

End Sub
```

Task: Use breakpoints and error handling to fix this.

Code 3 – Logic Error

```
Sub AverageMarks()

    Dim sum As Integer, count As Integer, i As Integer

    For i = 1 To 5

        sum = sum + Cells(i, 1).Value

    Next i
```

```
    MsgBox "Average: " & sum
End Sub
```

Task: Identify why the output is incorrect and correct it.

Note:

- Read the error message carefully — it often tells you the issue.
- Use **F8** slowly — watch variable changes.
- Keep variables meaningful (avoid “x”, “y” for everything).
- Use `Debug.Print` to display values in Immediate Window.
- Think logically — “What is the code *supposed* to do?”