Excel Assignment - 20

1. Write a VBA code to select the cells from A5 to C10. Give it a name “Data Analytics” and fill the cells with the following cells “This is Excel VBA”

Number	Odd or even
56	
89	
26	
36	
75	
48	
92	
58	
13	
25	

Ans-

```
Sub FillAndNameRange()
    Dim ws As Worksheet
    Dim dataRange As Range

    ' Define the worksheet (replace "Sheet1" with the name of your worksheet)
    Set ws = ThisWorkbook.Sheets("Sheet1")

    ' Define the range from A5 to C10
    Set dataRange = ws.Range("A5:C10")

    ' Name the range as "Data Analytics"
    ws.Names.Add Name:="DataAnalytics", RefersTo:=dataRange
```

```
    ' Fill the cells with the provided data
    dataRange.Cells(1, 1).Value = "This is Excel VBA"
    dataRange.Cells(2, 1).Value = "Number"
    dataRange.Cells(3, 1).Value = "Odd or Even"
    dataRange.Cells(4, 1).Value = 56
    dataRange.Cells(5, 1).Value = 89
    dataRange.Cells(6, 1).Value = 26
    dataRange.Cells(7, 1).Value = 36
    dataRange.Cells(8, 1).Value = 75
    dataRange.Cells(9, 1).Value = 48
    dataRange.Cells(10, 1).Value = 92
    dataRange.Cells(11, 1).Value = 58
    dataRange.Cells(12, 1).Value = 13
    dataRange.Cells(13, 1).Value = 25
End Sub
```

2. Use the above data and write a VBA code using the following statements to display in the next column if the number is odd or even

a. IF ELSE statement
b. Select Case statement
c. For Next Statement

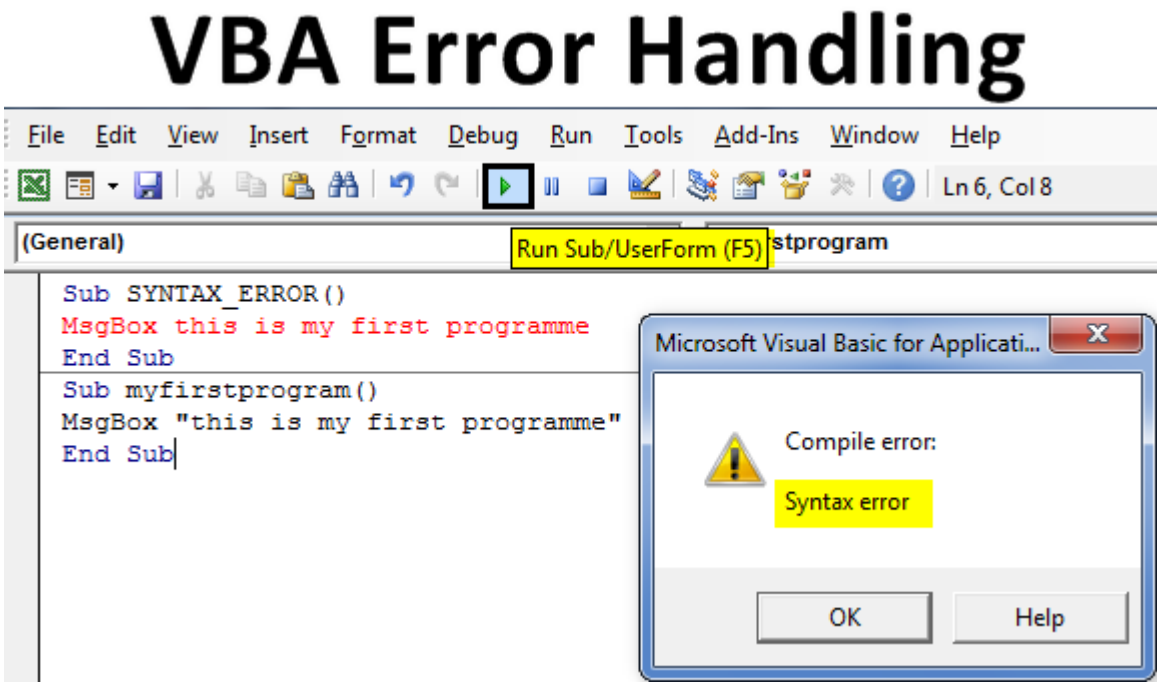
Ans-

```
Sub CheckOddEven()
    Dim ws As Worksheet
    Dim dataRange As Range
    Dim cell As Range

    ' Define the worksheet (replace "Sheet1" with the name of your worksheet)
    Set ws = ThisWorkbook.Sheets("Sheet1")

    ' Define the range from A5 to C10
    Set dataRange = ws.Range("A5:C10")

    ' Loop through each cell in the range
    For Each cell In dataRange
        If IsNumeric(cell.Value) Then
            ' Check if the cell value is numeric
            If cell.Value Mod 2 = 0 Then
                ' If it's even, display "Even" in the next column
                cell.Offset(0, 1).Value = "Even"
            Else
                ' If it's odd, display "Odd" in the next column
                cell.Offset(0, 1).Value = "Odd"
            End If
        End If
    Next cell
End Sub
```

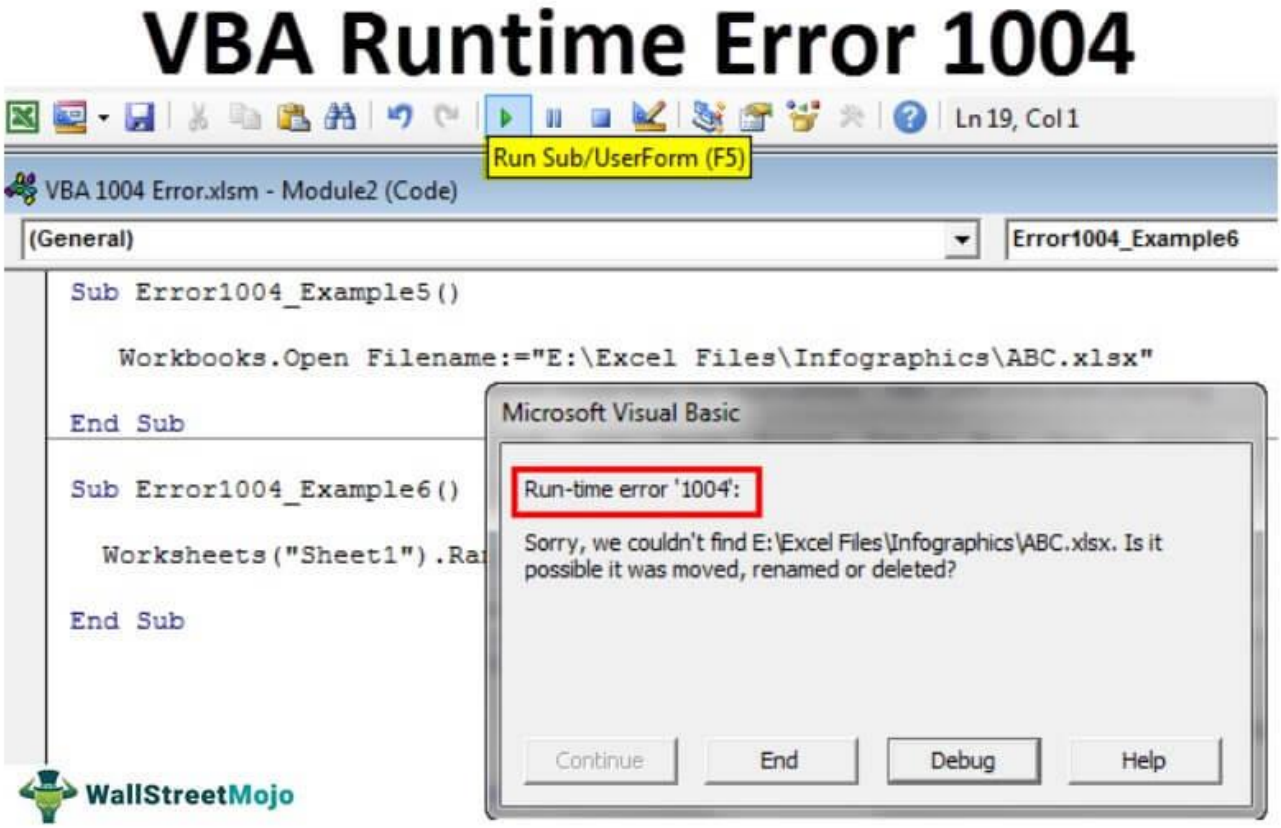
3. What are the types of errors that you usually see in VBA?

Ans-There are 3 main types of errors

a)Syntax Errors: occur during compile time



b)Runtime errors: occurs during runtime



c)Logical Errors: they do not cause any error but lead to

unexpected behaviour in programme.

4. How do you handle Runtime errors in VBA?

Ans- There are multiple techniques

a)On Error Resume Next Level: This is placed at the

beginning of code

On Error Resume Next

```
' Code that may cause an error

On Error GoTo 0 ' Reset error handling to default behavior

If Err.Number <> 0 Then

    ' Handle the error

    MsgBox "Error: " & Err.Description

    Err.Clear ' Clear the error object

End If
```

b)**ON Error GoTo Label:** directs VBA to jump to a

specified label

c)**On Error GoTO 0:** resets error handling to its default

behaviour

5. Write some good practices to be followed by VBA users for handling errors

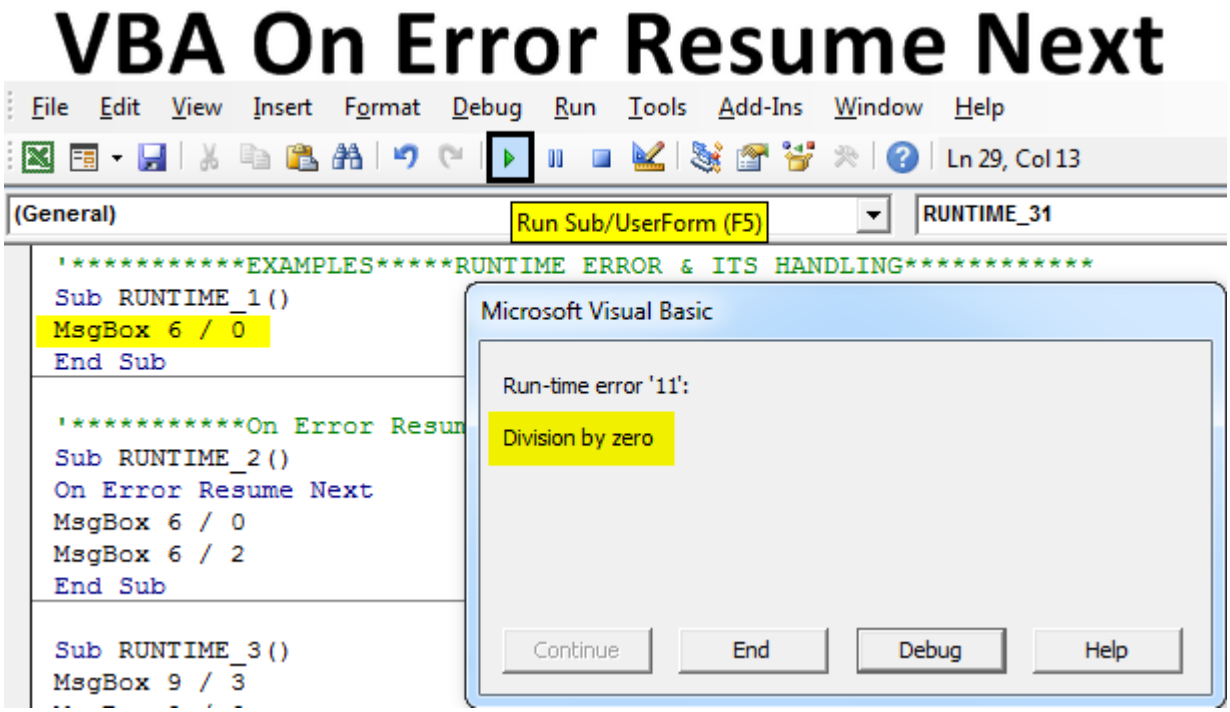
Ans- There are several meaningful practices such as

a)**Use Meaningful Error Messages:** provides clear and informative error messages

b)**Implement Error Handling Routines:**

c)Use Err Object

d)Clear Error Objects



6. What is UDF? Why are UDF's used? Create a UDF to multiply 2 numbers in VBA

Ans- UDF refers to User Defined Functions which is a custime fnction created by user to perform specific calculations or operations in excel.

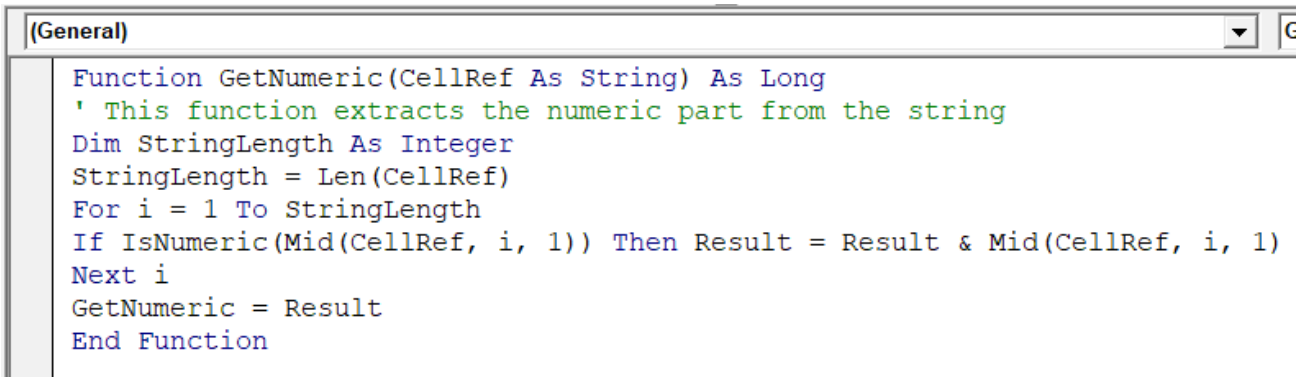
UDF Usages

a)Custom Functionality

b)Reusability

c)Clarity and documentation

d)Automation



UDF to multiple two numbers in VBA

```
Function MultiplyNumbers(ByVal num1 As Double, ByVal num2 As Double)
    As Double
    ' This function takes two numbers as input and returns their product
    MultiplyNumbers = num1 * num2
End Function
```