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”

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

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

4. How do you handle Runtime errors in VBA?

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

errors

Number Odd or even

56

89

26

36

75

48

92

58

13

25

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

SOLUTIONS:

1. Select and Name Cells:

```
Sub SelectAndNameCells()
    ' Select cells from A5 to C10
    Range("A5:C10").Select

    ' Give the selected range a name "DataAnalytics"
    Selection.Name = "DataAnalytics"

    ' Fill the cells with the text "This is Excel VBA"
    Selection.Value = "This is Excel VBA"
End Sub
```

2. Display Odd or Even using Different Statements:

```
Sub DisplayOddEven()

  Dim cell As Range

' Loop through each cell in the named range "DataAnalytics"

    For Each cell In Range("DataAnalytics")

   ' Using IF ELSE statement

     If cell.Value Mod 2 = 0 Then

     cell.Offset(0, 3).Value = "Even"

     Else
```

```
    cell.Offset(0, 3).Value = "Odd"
    End If

    ' Using Select Case statement
    Select Case cell.Value Mod 2
    Case 0
        cell.Offset(0, 4).Value = "Even"
    Case 1
    cell.Offset(0, 4).Value = "Odd"
  End Select
' Using For Next statement
    For i = 5 To 9
    If cell.Value Mod 2 = 0 Then
      cell.Offset(0, i).Value = "Even"
    Else
        cell.Offset(0, i).Value = "Odd"
    End If
 Next i
    Next cell
End Sub
```

3. Types of Errors in VBA:

- **Syntax Errors:** Mistakes in the code structure.
- **Runtime Errors:** Occur during code execution (e.g., division by zero).
- **Logic Errors:** Code runs but produces incorrect results.

4. Handling Runtime Errors in VBA:

- Use error handling with **On Error Resume Next** or **On Error GoTo [label]**.
- Identify and fix the root cause of the error.
- Use **Err.Number** and **Err.Description** to get information about the error.

5. Good Practices for Handling Errors:

- Always use **Option Explicit** to force variable declaration.
- Use meaningful variable and procedure names.
- Include comments for complex or critical parts of the code.
- Test your code thoroughly, including boundary cases.
- Use error handling to gracefully manage unexpected situations.

6. User-Defined Function (UDF) to Multiply 2 Numbers:

```
Function MultiplyNumbers(num1 As Double, num2 As Double) As Double

    ' UDF to multiply two numbers

    MultiplyNumbers = num1 * num2

End Function
```

You can use this UDF in a cell like =MultiplyNumbers(A1, B1) to multiply the values in cells A1 and B1.