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”

Ans

```
Sub DataAnalytics()

  Range("A5:C10").Select

  ActiveWorkbook.Names.AddName:="Data_Analytics",RefersTo:=Selection

  Selection.Value = "This is Excel VBA"

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

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

Ans:

a. IF ELSE statement

```
 CheckOddEven_IfElse()
    Dim ws As Worksheet
    Dim targetRange As Range
    Dim cell As Range

    Set ws = ThisWorkbook.Sheets("Sheet1")
    Set targetRange = ws.Range("I2:I11")

    For Each cell In targetRange
        If cell.Value Mod 2 = 0 Then
            cell.Offset(0, 1).Value = "Even"
        Else
            cell.Offset(0, 1).Value = "Odd"
        End If
    Next cell
End Sub
```

b. Select Case statement

```
Sub CheckOddEven_SelectCase()
    Dim ws As Worksheet
    Dim targetRange As Range
    Dim cell As Range

    Set ws = ThisWorkbook.Sheets("Sheet1")
    Set targetRange = ws.Range("I2:I11")

    For Each cell In targetRange
        Select Case cell.Value Mod 2
            Case 0
                cell.Offset(0, 1).Value = "Even"
            Case 1
                cell.Offset(0, 1).Value = "Odd"
        End Select
    Next cell
End Sub
```

c. For Next Statement

```
Sub CheckOddEven_ForNext()
    Dim ws As Worksheet
    Dim targetRange As Range
    Dim cell As Range
    Dim i As Long
    Set ws = ThisWorkbook.Sheets("Sheet1")
    Set targetRange = ws.Range("I2:I11")

    For i = 1 To targetRange.Rows.Count
        If targetRange.Cells(i, 1).Value Mod 2 = 0 Then
            targetRange.Cells(i, 1).Offset(0, 1).Value = "Even"
        Else
            targetRange.Cells(i, 1).Offset(0, 1).Value = "Odd"
        End If
    Next i
End Sub
```

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

Ans:

1. Syntax errors – It occurs when a specific line of code is not written correctly.

2. **Compile errors** – It happens when issues happen when putting together lines of code, though the individual lines of code seem to make sense.

3. **Runtime errors** – Runtime Error occurs when the code is usually correct in principle, but an action taken by the user, or the data being used leads to unexpected errors.

4. How do you handle Runtime errors in VBA?

Ans:

The primary tools for error handling in VBA are 'On Error' statement and 'Err' object.

a. The On Error statement allows you to specify how VBA should handle errors. There are 3 options –
    i) On Error Resume Next
    ii) On Error GoTo 0
    iii) On Error GoTo Label

b. The Err object provides information about the most recent runtime error that occurred.

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

Ans:

a. Use ‘explicit’ option at top of your models.

b. Include appropriate error handling statements.

c. Reset error handlings to its default state.

d. Log errors to file or a log sheet.

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

Ans:

UDF stands for User Defined Function. In VBA, it is used for performing a specific task.

a. UDF’s can be used repeatedly in different parts of workbooks promoting code reusability.
b. UDF’s cab enhance readability of your formulas by encapsulating complex calculations.
c. UDF’s can automate specialized tasks making your worksheets more efficient.

Code:

```
Function MultiplyNumbers(ByVal num1 As Double, ByVal num2 As Double) As Double
    MultiplyNumbers = num1 * num2
End Function
```