

**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 |  |

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



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

**ANS:**

1. VBA code to select cells and fill with data:

```vba

Sub SelectCellsAndFill()

Range("A5:C10").Name = "DataAnalytics"

Range("DataAnalytics").Value = "This is Excel VBA"

End Sub

```

2. VBA code to display if numbers are odd or even:

a. IF ELSE statement:

```vba

Sub OddEven\_IFELSE()

Dim cell As Range

For Each cell In Range("DataAnalytics").Columns(1).Cells

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:

```vba

Sub OddEven\_SelectCase()

Dim cell As Range

For Each cell In Range("DataAnalytics").Columns(1).Cells

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:

```vba

Sub OddEven\_ForNext()

Dim i As Long

For i = 1 To 6 ' Looping through rows 5 to 10

If Cells(i + 4, 1).Value Mod 2 = 0 Then

Cells(i + 4, 2).Value = "Even"

Else

Cells(i + 4, 2).Value = "Odd"

End If

Next i

End Sub

```

3. Types of errors in VBA:

- Syntax errors

- Runtime errors

- Logic errors

4. Handling Runtime errors in VBA:

- Using error handling techniques such as `On Error Resume Next`, `On Error GoTo`, and `On Error GoTo 0`.

- Using `Err` object to capture error details.

- Properly handling exceptions to avoid unexpected behavior or crashes.

5. Good practices for handling errors in VBA:

- Use error handling to anticipate and gracefully manage errors.

- Provide informative error messages to aid debugging and troubleshooting.

- Use proper testing and debugging techniques to catch errors early in the development process.

- Document error handling strategies in the code for future reference.

6. UDF (User Defined Function):

- UDFs are custom functions created by the user to perform specific tasks in Excel.

- UDFs are used to extend the functionality of Excel by providing custom calculations or operations.

- Here's a UDF to multiply two numbers in VBA:

```vba

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

MultiplyNumbers = num1 \* num2

End Function

```

To use this UDF in Excel, you can enter `=MultiplyNumbers(A1, B1)` in a cell, where A1 and B1 are the cells containing the numbers you want to multiply.