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:- VBA code to select cells and fill with data:

```
Sub SelectCells()
   Range("A5:C10").Select
   Selection.Name = "Data Analytics"
   Range("A5:C10").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

ANS:- VBA code using different statements to display if the number is odd or even:

a. <u>IF ELSE statement</u>:

```
Sub CheckOddEven()
   Dim cell As Range

   For Each cell In Range("D2:D10")
      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. <u>Select Case statement</u>:

```
Sub CheckOddEven()
```

```
  Dim cell As Range

  For Each cell In Range("D2:D10")
    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()
  Dim i As Integer

  For i = 2 To 10
    If Cells(i, 4).Value Mod 2 = 0 Then
      Cells(i, 5).Value = "Even"
    Else
      Cells(i, 5).Value = "Odd"
    End If
  Next i
End Sub
```

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

ANS:- The types of errors that you may encounter in VBA are syntax errors, runtime errors, and logical errors.

4. How do you handle Runtime errors in VBA?

ANS:- To handle runtime errors in VBA, you can use error handling techniques such as On Error statement, which allows you to catch and handle errors in your code. You can also use the Err object to obtain information about the error that occurred.

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

Errors

ANS:- Some good practices to follow for handling errors in VBA include always using error handling techniques, such as the On Error statement, testing your code thoroughly before deploying it, and using descriptive names for your variables and functions to make it easier to identify errors. It is also important to document your code with comments and provide clear error messages for users.

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

ANS:- UDF stands for User-Defined Function. It is a function that is created by the user in VBA and can be used in Excel formulas. UDFs are used to extend the functionality of Excel beyond the built-in functions.

To create a UDF to multiply 2 numbers in VBA, follow the below steps:

Open VBA Editor by pressing ALT + F11.

Click on Insert > Module to create a new module.

Write the following code to create a UDF named "MultiplyNumbers":

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

Save the code and close the VBA Editor.

Go to Excel and enter the formula "=MultiplyNumbers(A1,B1)" in any cell to multiply the values in cells A1 and B1.

This UDF will take two arguments (num1 and num2) and return their product. It can be used in Excel formulas just like any other built-in function.