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”  
  
Sub SelectRange()  
Dim Data\_Analytics As Range  
Set Data\_Analytics = Range("A5:C10").Select  
Range(Data\_Analytics).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

**IF ELSE statement:**   
  
Sub evenOdd()

If Range("A2") Mod 2 = 0 Then

Range("B2").Value = "even"

Else

Range("B2").Value = "odd"

End If

End Sub

**Select Case statement**

Sub evencheck()

Select Case Range("A2") Mod 2

Case Is = 0

Range("B2").Value = "even"

Case Else

Range("B2").Value = "odd"

End Select

End Sub

**For Next Statement**

Sub even\_check()

Dim LastRow As Long

LastRow = Sheet1.Cells(Sheet1.Rows.Count, 1).End(xlUp).Row

Dim i As Long, Total As Long

For i = 2 To LastRow

If Range("A2") Mod 2 = 0 Then

Range("B2").Value = "even"

Else

Range("B2").Value = "odd"

End If

Next i

End Sub

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

* Syntax errors.
* Compilation errors.
* Runtime errors.
* Logical Errors.

4. How do you handle Runtime errors in VBA?

We can enable the debug option and check there the error occurred. It can be enabled using below: Options> General> Error> Break on Unhandled Errors  
1. Error Trapping – We can think possible way of invalid input and write code to handle such issues.   
2. Clear User Instructions – It prevents users to provide invalid input

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

Good practices to be followed by VBA users for handling errors:

* Ignore the error and let the code continue
* Have an error handling code in place and run it when an error occurs.

On Error statements helps to handle error in better way:

On Error Resume Next  
On Error GoTo 0  
On Error Goto [Label]  
On Error Goto -1

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

UDF is helps in creating custom functions and helps in extending the system to perform the operations that are not available with inbuilt functions. User-defined functions help you encapsulate and reuse functionality.

Function Multiply(x as Integer y As Integer ) As Integer  
Z = x\*y  
End Function