**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”

Answer:

Sub Data\_Analytics()

Dim string\_name As String

string\_name = "This is Excel VBA"

ActiveSheet.Range("A5:C10").Select

Range("A5:C10") = string\_name

End Sub

2. Use the 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

Answer:

With For Next Statement

Sub OddEven()

For i = 2 To 11

a = Cells(i, 1).Value

b = a Mod 2

If b = 0 Then

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

Else

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

End If

Next

End Sub

With Select Case Statement

Sub OddEven\_Select()

For i = 2 To 11

a = Cells(i, 1).Value

b = a Mod 2

Select Case b

Case Is = 0

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

Case Is <> 0

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

End Select

Next

End Sub

A screenshot of a computer

Description automatically generated

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

Answer: Types of error in VBA

* Syntax error- occurs when VBA finds something wrong with syntax in code.
* Compile error-occurs when something is missing that is needed for the code to run.
* Run Time Error-occurs when code is running
* Logical Error- this error would not make code stop but lead to wrong results. These could also be the most difficult types of errors to troubleshoot.

4. How do you handle Runtime errors in VBA?

Answer: Runtime errors are those that appear only after you compile and run your code. For example: you might correctly write a line of code to open the file. However, if that file does not exist, application can not open the file. Then it can throw the error.

When A runtime error occurs, it will stop the code and show you the error dialog box. The message in dialog box will help you to understand the problem and correct it. When you click on debug button, it will highlight the part of the code that is leading to error. When error is fixed, you can click on run button in toolbar to continue running the code from where it left, or you can click on end button to come out from the code.

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

Answer: Here are some best practices you can use when it comes to error handling in Excel VBA.

* Use ‘On Error Go [Label]’ at the beginning of the code. This will make sure any error that can happen from there is handled.
* Use ‘On Error Resume Next’ ONLY when you’re sure about the errors that can occur. Use it with expected error only. In case you use it with unexpected errors, it will simply ignore it and move forward. You can use ‘On Error Resume Next’ with ‘Err.Raise’ if you want to ignore a certain type of error and catch the rest.
* When using error handlers, make sure you’re using Exit Sub before the handlers. This will ensure that the error handler code is executed only when there is an error (else it will always be executed).
* Use multiple error handlers to trap different kinds of errors. Having multiple error handler ensures that an error is properly addressed. For example, you would want to handle a ‘type mismatch’ error differently than a ‘Division by 0’ run-time error.

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

Answer: UDF is User defined function. These are custom functions which we create to use them in worksheet to cater specific needs.

We use UDF when existing functions are not enough to solve the task. Sometimes, When we work with formulas, it is hard to read and understand by others. Custom formula will make it easier as you don’t need to write the complex formula again and again. Moreover, when VBA code is converted into function that you do not need to run code again and again. we can simply insert it as function.