**Advance Excel Assignment 20**

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

Ans. In Visual Basic, errors fall into one of three categories: syntax errors, run-time errors, and logic errors :

**Syntax Errors**

Syntax errors are those that appear while you write code. If you're using Visual Studio, Visual Basic checks your code as you type it in the Code Editor window and alerts you if you make a mistake, such as misspelling a word or using a language element improperly. If you compile from the command line, Visual Basic displays a compiler error with information about the syntax error. Syntax errors are the most common type of errors. You can fix them easily in the coding environment as soon as they occur.

**Run-Time Errors**

Run-time errors are those that appear only after you compile and run your code. These involve code that may appear to be correct in that it has no syntax errors, but that will not execute. For example, you might correctly write a line of code to open a file. But if the file does not exist, the application cannot open the file, and it throws an exception. You can fix most run-time errors by rewriting the faulty code or by using exception handling, and then recompiling and rerunning it.

**Logic Errors**

Logic errors are those that appear once the application is in use. They are most often faulty assumptions made by the developer, or unwanted or unexpected results in response to user actions. For example, a mistyped key might provide incorrect information to a method, or you may assume that a valid value is always supplied to a method when that is not the case. Although logic errors can be handled by using exception handling, most commonly they should be addressed by correcting the error in logic and recompiling the application.

1. How do you handle Runtime errors in VBA?

Ans. To handle an error inline, use the Resume Next statement with On Error. Any errors that occur during runtime cause Info Connect to continue executing the macro at the next statement. If an error occurs, it is handled by opening a dialog box, passing control to another procedure or to a routine within the same procedure.

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

Ans.

|  |  |
| --- | --- |
| **Description** | **VBA Code** |
| On Error – Stop code and display error | On Error Goto 0 |
| On Error – Skip error and continue running | On Error Resume Next |
| On Error – Go to a line of code [Label] | On Error Goto [Label] |
| Clears (Resets) Error | On Error GoTo –1 |
| Show Error number | MsgBox Err.Number |
| Show Description of error | MsgBox Err.Description |
| Function to generate own error | Err.Raise |

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

Ans. A user-defined function (UDF) is a common fixture in programming languages, and the main tool of programmers for creating applications with reusable code. Since programs are mostly composed of code that comes from the programmer, or in this case the user, most of it is composed of user-defined functions occasionally punctuated by built-in functions.

User-defined functions allow programmers to create their own routines and procedures that the computer can follow; it is the basic building block of any program and also very important for modularity and code reuse since a programmer could create a user-defined function which does a specific process and simply call it every time it is needed. Their syntax depends entirely on the programming language or application where they are created.

Function Mul()  
  
ActiveCell.Offset(0, 115).Range("A1").Select  
ActiveCell.Offset(0, -115).Range("A1").Select  
ActiveCell.FormulaR1C1 = "=RC[11]\*RC[115]"  
ActiveCell.Offset(1, 0).Range("A1").Select  
End Function

1. When is offset statement used for in VBA? Let’s suppose your current highlight cell is A1 in the below table. Using OFFSET statement, write a VBA code to highlight the cell with “Hello” written in it.

A B C

1 25 354 362

2 36 6897 962

3 85 85 Hello

4 96 365 56

5 75 62 2662

Ans. Range ("A1").Select

Range("RangeName").Select

Cells(3, 4).Select 'Selects Row 3, Column 4, i.e. cell D3

For i = 1 to 10

Cells(i, 1).value = i ' fill A1 through A10 with the value of i

Next i

1. What are the data types used in VBA?

Ans. Below is a list of common VBA variables (known as data types) used in macros and their purposes:

* Integer: Used to store number values that won’t take on decimal form.
* Single: Used to store number values that may take on decimal form. Can also contain integers.
* Double: A longer form of the single variable. Takes up more space, but needed for larger numbers.
* Date: Stores date values.
* String: Stores text. Can contain numbers, but will store them as a text (calculations cannot be performed on numbers stored as a string)
* Boolean: Used to store binary results (True/False, 1/0)