VBA (VISUAL BASIC APPLICATION)

VBA (Visual Basic for Applications) is the programming language of Excel and other Office programs. 1 Create a Macro: With Excel VBA you can automate tasks in Excel by writing so called macros. A loop in Excel VBA enables you to loop through a range of cells with just a few codes lines.

VBA is a true-Object Oriented programing Language.
Visual: - Transparency
Basic: - Beginner All Purpose Symbolic Instruction Code
Application: - Excel, Word, Power Point, Outlook
Language: - Mode of communication
Program: - Program is a set of Instructions
Programing Language: - Programing language is systematic way to execute the program.

Types of programing Language

a) Low level Programing Language	Assembling Language Ex. Windows, Dos, Linux, Unix
b) High Level Programing Language	Developing Language Ex. Vb, java, c, c++, Cobol, Pascal, C#, Avap etc.
c) Application based programing Language	VBA-Excel, **VBA-Access**, VBA-Outlook, VBA-Word, VBA-PowerPoint, VBA SQL

Venue to Write Program	
1) Workbook	Use for All worksheet of workbook
2) Worksheet	use for particular worksheet
3). Module	a) Standard Module
	b) Form Module
	c) Class Module

By *Gunjan Kumar Gupta*

VBA (VISUAL BASIC APPLICATION)

Class	Class is a set of Object/Collections of Object	Ex. Workbook, Worksheet, Range, Cells, Row, Column, Pivot table
Object	Object is an instance/example of Class	Ex. Book1, Sheet1, Range("A1:A10"), Cells(1,1)
Property	Property is an attribute of object	ex. Color, size, height, width, length
Procedure	Subroutine Procedure	
	Function Procedure	
	Property Procedure	
Event	Event is an action which performed as per our need.	Ex. Open, close, Save, Activate, Change, Click, Double Click etc

NAMING CONVENTION

Rule for Naming Convention:

1 Name Should be alpha Numeric or Should Alpha
ex. Abc123, abc
Invalid: 123abc, 123

2 Avoid Space Between word
ex. My Program

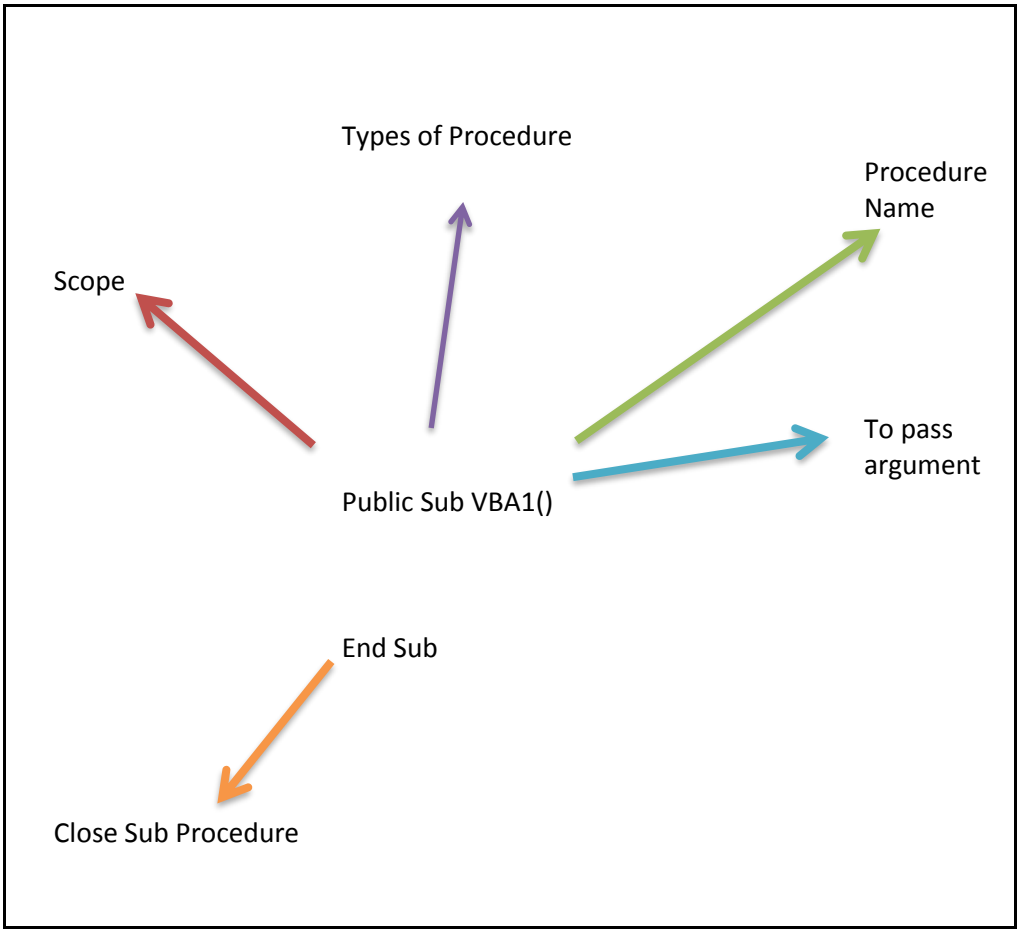
3 Avoid Special Characters except _(Underscore)

4 Avoid System defined/Keyword
ex. Sub, Name, Public, Private

5 Not more than 255 Characters

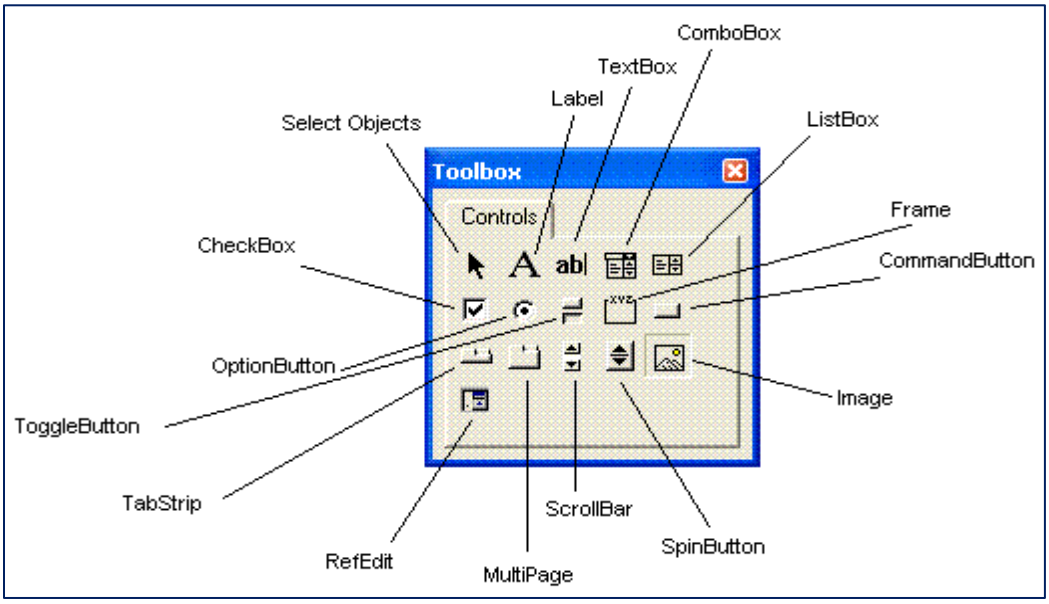
By Gunjan Kumar Gupta

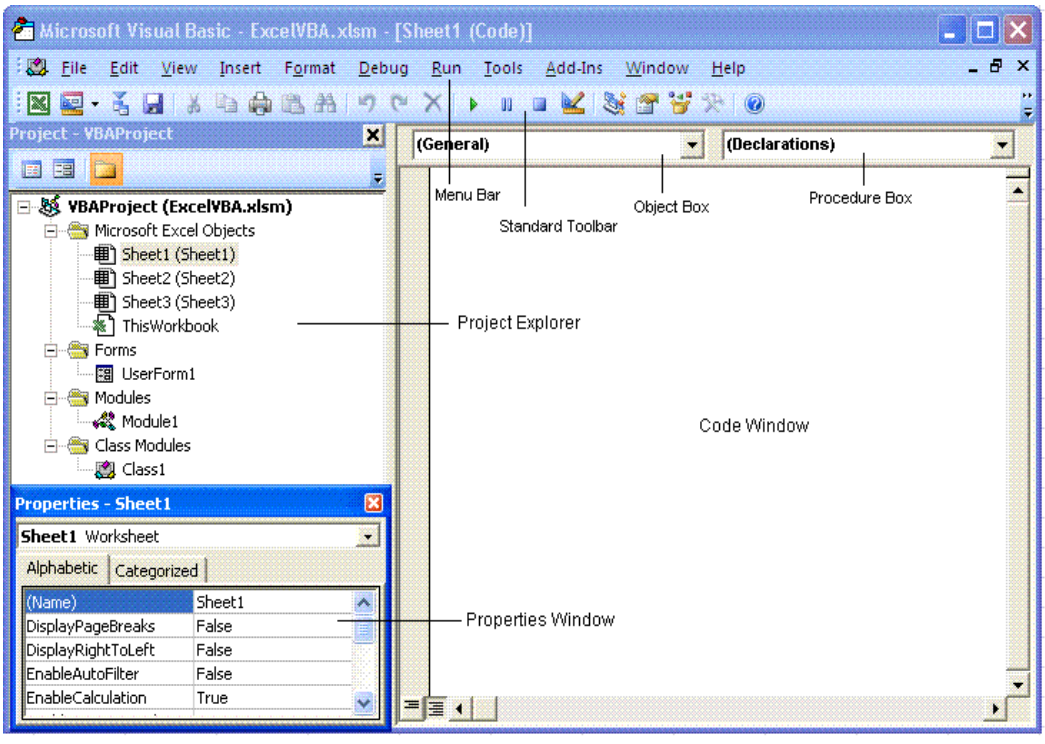
VBA (VISUAL BASIC APPLICATION)



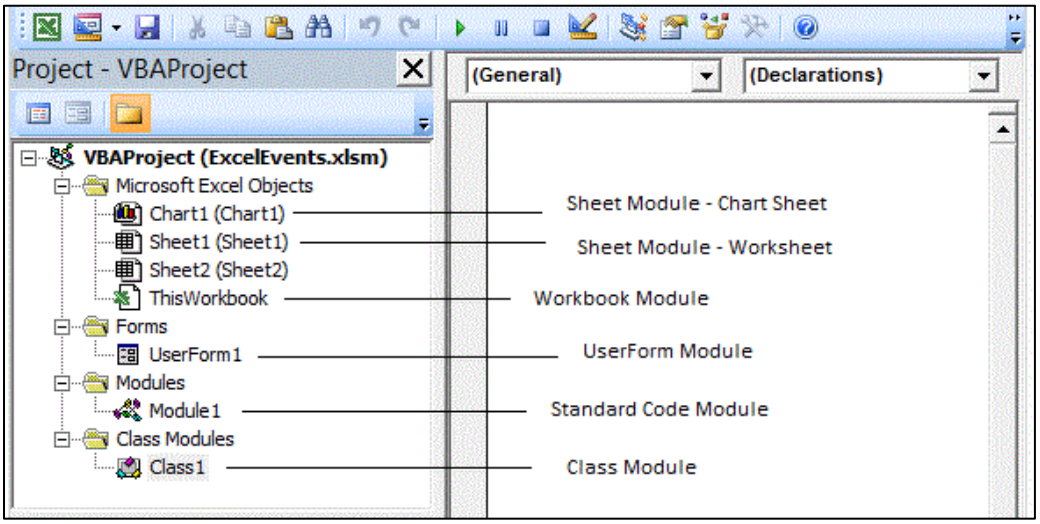
Data Type (VARIABLES)			OBJECT	
Excel	**VBA**	**Type-Declaration Character/Suffixes**	**Object**	**Class**
Number	Integer	%	rng	Range
Number	Long	&	sh	Worksheet, Sheet
Decimal	Single	!	wkb	workbook, Thisworkbook
Decimal	Double	#	tkb/akb	workbook, Thisworkbook
Text	String	@	pvt	Pivot table
Currency	Currency	$	chrt	Chart
			Obj	Object

VBA (Visual Basic Application)





VBA (VISUAL BASIC APPLICATION)



Macro	VBA
ABSOLUTE MACRO	**RELATIVE MACRO**
Sub Micro1() ' ' ABC Macro ' ' Range(Selection, Selection.End(xlDown)).Select **Selection.Copy** Range("D1").Select **ActiveSheet.Paste** **End Sub**	Sub VBA1() ' ' ABC Macro ' ' Range(Selection, Selection.End(xlDown)).Select **Selection.Copy** ActiveCell.Offset(0, 2).Range("A1").Select **ActiveSheet.Paste** **End Sub**

Interview Question: -

1	Why we Use Option Explicit?
Ans	Option Explicit use to declare variable/Object forcefully.
2	Why we declare Variable/Object?

Ans	we can create temp data store during execution period.
3	Why we use Data Type.
Ans	1)To optimization of Compiler's Memory. 2)To use of correct execution of Program
4	Types of Macro
Ans	1) Record Macro 2) Run Macro(VBA)
5	Difference between VBA and Macros 1) VBA is conditional but Macro is condition less 2) Macro can convert in VBA but vba can't convert in Record Macro 3) Vba is dynamic but macro is not dynamic
6	Difference between VB, VBA, VBS * VB is stand alone program that runs independently * VBA is part of excel program and can not work alone * VBS is variant of Visual Basic Language used for Internet Application

SL. No.	Topic	Covered
Q-1	Count All Sheets	

```
Sub Q1()
        'MsgBox Sheets.Count
        MsgBox "Total No of Sheets:- " & Sheets.Count
End Sub
```

SL .No.	Topic	Covered
Q-2	Add A Sheet	

```
Sub Q2()
        'Sheets.Add
        Sheets.Add after:=Sheets(Sheets.Count)
End Sub
```

SL. No.	Topic	Covered
Q-3	Delete A Sheet	

```
Sub Q3()
        Sheet5.Delete
```

End Sub

Discerption:

Sheet5 is the name of sheet of worksheet which will be delete.

SL. No.	Topic	Covered
Q-4	Delete Multiple Sheet	

```
Sub Q4()
    'Sheets(Array("sheet4", "sheet2", "sheet3")).Delete
    Sheets(Array(2, 3)).Delete
End Sub
```

Discerption:

This code will delete all sheets which is passed on Array.

SL. No.	Topic	Covered
Q-5	Rename A Sheet	

```
Sub Q5()
    Sheet1.Name = "Gunjan"
End Sub
```

SL. No.	Topic	Covered
Q-6	Hide A Sheet	

```
Sub Q6()
    'Sheet9.Visible = xlSheetHidden
    Sheet10.Visible = xlSheetVeryHidden
End Sub
```

SL. No.	Topic	Covered
Q-7	Unhide A Sheet	

```
Sub Q7()
    Sheet10.Visible = xlSheetVisible
```

```
End Sub
```

SL. No.	Topic	Covered
Q-8	Move Within Workbook	

```
Sub Q8()
    'Sheet1.Move after:=Sheets(Sheets.Count)
    Sheet1.Move after:=Sheet10
End Sub
```

SL. No.	Topic	Covered
Q-9	Move Within Another Workbook	

```
Sub Q9()
    Dim wkb As Workbook
    Set wkb = Workbooks("Book2")
    Sheet1.Copy after:=wkb.Sheets(Sheets.Count)
End Sub
```

SL. No.	Topic	Covered
Q-10	Color Sheet Tab	

```
Sub Q10()
    Sheet1.Tab.ColorIndex = 22
End Sub
```

Discerption:

We can put color Code between -> **0-255.**

SL. No.	Topic	Covered
Q-11	Copy A Sheet	

```
Sub Q11()
    Sheet1.Copy after:=Sheets(Sheets.Count)
```

```
End Sub
```

SL. No.	Topic	Covered
Q-12	Protect A Sheet	

```
Sub Q12()
    ActiveSheet.Protect "Abc"
End Sub
```

SL. No.	Topic	Covered
Q-13	Unprotect A Sheet	

```
Sub Q13()
    ActiveSheet.Unprotect "Abc"
End Sub
```

SL. No.	Topic	Covered
Q-14	Protect A Range	

```
Sub Q14()
    Dim sh As Worksheet
    Dim rng As Range
    Dim i As Integer
    Set sh = ThisWorkbook.Sheets(1)
        i = sh.Cells(Rows.Count, "C").End(xlUp).Row
        Set rng = sh.Range("A1:C" & i)
    Cells.Select
    sh.Unprotect "abc"
    Selection.Locked = False
    rng.Locked = True
    sh.Protect "abc"
End Sub
```

SL. No.	Topic	Covered
Q-15	Print Particulars Sheet	

```
Sub Q15()
    Dim akb As Workbook
```

```
    Set akb = ActiveWorkbook
    akb.PrintOut
    'Application.Dialogs(xlDialogPrintPreview).Show
    Application.Dialogs(xlDialogPrinterSetup).Show
End Sub
```

SL. No.	Topic	Covered
Q-16	Select Current Sheet	

```
Sub Q16()
    Cells.Select
End Sub
```

SL. No.	Topic	Covered
Q-17	Open A Workbook	

```
Sub Q17()
  Dim str As String
    str = VBA.InputBox("Enter URL")
    Workbooks.Open Filename:=str
End Sub
```

SL. No.	Topic	Covered
Q-18	New Workbook	

```
Sub Q18()
    Workbooks.Add
End Sub
```

SL. No.	Topic	Covered
Q-19	Save As A Workbook	

```
Sub Q19()
```

```
    ActiveWorkbook.SaveAs Filename:="C:\Users\Nilesh\Desktop\VBA-17th
    Oct'19\demo.xlsm"
End Sub
```

SL. No.	Topic	Covered
Q-20	Save Workbook	

```
Sub Q20()
    ActiveWorkbook.Save
End Sub
```

SL. No.	Topic	Covered
Q-21	Protect A Workbook	

```
Sub Q21()
    ActiveWorkbook.Password = demo
    ActiveWorkbook.Save
End Sub
```

SL. No.	Topic	Covered
Q-22	Close Workbook	

```
Sub Q22()
    ActiveWorkbook.Close
End Sub
```

SL. No.	Topic	Covered
Q-23	Close Application	

```
Sub Q23()
    Application.Quit
End Sub
```

SL. No.	Topic	Covered
Q-24	Delete File	

```
Sub Q24()
```

```
Kill "C:\Users\Nilesh\Desktop\Dashboard\*.xls"
End Sub
```

SL. No.	Topic	Covered
Q-25	Delete All File Of A Folder	

```
Sub Q25()
Kill "C:\Users\Nilesh\Desktop\Dashboard\*."
End Sub
```

SL. No.	Topic	Covered
Q-26	Show Workbook Name	

```
Sub Q26()
Range("F3").Value = (ActiveWorkbook.Name)
End Sub
```

3rd Class

SL. No	Topic	Covered
Q-1	Select A Cell On The Active Worksheet	**Red**

```
Sub Q1()
        ActiveCell.Interior.Color = vbRed
End Sub
```

SL. No	Topic	Covered
Q-2	Select A Cell On The Another Sheet In The Active Workbook	**Green**

```
Sub Q2()
        Sheet2.Range("b7").Interior.Color = vbGreen
        End Sub
```

SL. No	Topic	Covered
Q-3	Select A Cell On The Active Worksheet In Different Workbook	

```
Sub Q3()
```

```
    Dim wkb As Workbook
    Set wkb = Workbooks("Book2")
    wkb.Activate
    Range("c5").Interior.Color = vbYellow
End Sub
```

SL. No	Topic	Covered
Q-4	Select A Range On The Active Worksheet	

```
Sub Q4()
    Range("b4:D7").Interior.Color = vbBlue
End Sub
```

SL. No	Topic	Covered
Q-5	Select A Range On The Another Sheet In The Active Workbook	

```
Sub Q5()
    Sheet2.Range("b7:O9").Interior.Color = vbGreen
End Sub
```

SL. No	Topic	Covered
Q-6	Select A Range On The Active Worksheet In Different Workbook	

```
Sub Q6()
    Dim wkb As Workbook
    Set wkb = Workbooks("Book2")
    wkb.Activate
    Range("c5:u8").Interior.Color = vbYellow
End Sub
```

SL. No	Topic	Covered
Q-7	Create Naming Range On The Active Worksheet	

```
Sub Q7()
    Range("G5:G10").Name = "Gunjan"
End Sub
```

SL No	Topic	Covered
Q-8	Create Naming Range On The Another Sheet In The Active Workbook	

```
Sub Q8()
```

```
        Sheet1.Next.Select
        Range("G5:G10").Name = "Gunjan"
End Sub
```

SL. No	Topic	Covered
Q-9	Create Naming Range On The Active Worksheet In Different Workbook	

```
Sub Q9()
Dim wkb As Workbook
Set wkb = Workbooks("Book2")
        wkb.Activate
        Range("G5:G10").Name = "Gunjan"
End Sub
```

SL. No	Topic	Covered
Q-10	Colored A Cell On The Active Worksheet But After Executing The Program Color Should Be Change.	

```
Sub Q10()
        Range("F10:F15").Interior.ColorIndex = Application.WorksheetFunction.RandBetween(1, 52)
End Sub
```

SL. No	Topic	Covered
Q-11	Colored A Cell On The Another Sheet In The Same Workbook But After Executing The Program Color Should Be Change.	

```
Sub Q11()
        Sheet1.Next.Select
        Range("F10:F15").Interior.ColorIndex = Application.WorksheetFunction.RandBetween(1, 52)
End Sub
```

SL. No	Topic	Covered
Q-12	Colored A Cell On The Active Sheet In Different Workbook But After Executing The Program Color Should Be Change	

```
Sub Q12()
Dim wkb As Workbook
Set wkb = Workbooks("Book2")
        wkb.Activate
        Range("F10:F15").Interior.ColorIndex = Application.WorksheetFunction.RandBetween(1, 52)
End Sub
```

SL. No	Topic	Covered
Q-13	Colored Cell Beside Of Active Cell (Eg. Left/Right/Up/Down) On The Active Worksheet	

```
Sub Q13()
        ActiveCell.Offset(1, 0).Interior.Color = vbRed
        ActiveCell.Offset(-1, 0).Interior.Color = vbGreen
        ActiveCell.Offset(0, 1).Interior.Color = vbBlue
        ActiveCell.Offset(0, -1).Interior.Color = vbYellow
End Sub
```

SL. No	Topic	Covered
Q-13	Colored Cell Beside Of Active Cell (Eg. Left/Right/Up/Down) On The Active Worksheet	
Q-14	Colored Cell Beside Of Active Cell (Eg. Left/Right/Up/Down) On The Another Worksheet In The Same Workbook	
Q-15	Colored Cell Beside Of Active Cell (Eg. Left/Right/Up/Down) On The Active Worksheet In The Defferent Workbook	

Q-13 & Q-14 Take help of previous help….

SL. No	Topic	Covered
Q-16	Colored Next Range of The Created Naming Range on The Active Worksheet	Blue
Q-17	Colored Next Range of The Created Naming Range on The Another Worksheet in The Active Workbook	
Q-18	Colored Next Range of The Created Naming Range on The Active Worksheet in The Different Workbook	

```
Sub Q16()
        Range("Gunjan").Offset(0, 1).Interior.Color = vbBlue
End Sub
```

SL. No	Topic	Covered
Q-19	Select The Last Cell of A Column Of Used Data	

```
Sub Q19()
```

```
    Dim i As Integer
    i = Sheet1.Cells(1, 1).End(xlDown).Row
    Sheet1.Range("B" & i).Select
End Sub
```

SL. No	Topic	Covered
Q-20	Select The Last Blank Cell Of A Column Of Used Data	

```
Sub Q20()
    Dim i As Integer
    i = Sheet1.Cells(1, 1).End(xlDown).Row + 1
    Sheet1.Range("A" & i).Select
End Sub
```

SL. No	Topic	Covered
Q-21	Last Used Row No Show In Msg Box Of A Column Of Used Data	

```
Sub Q21()
Dim i As Integer
    i = Sheet1.Cells(1, 1).End(xlDown).Row
    Sheet1.Range("B" & i).Select
    MsgBox "Last Used Rows:- " & i
End Sub
```

SL. No	Topic	Covered
Q-22	Last Blank Row No Show in Msg Box Of A Column Of Used Data	

```
Sub Q22()
    Dim i As Integer
    i = Sheet1.Cells(1, 1).End(xlDown).Row + 1
    Sheet1.Range("B" & i).Select
    MsgBox "Last Used Rows:- " & i
End Sub
```

SL. No	Topic	Covered
Q-23	Select all data in current region	

```
Sub Q23()
```

```
    Range("A1").CurrentRegion.Select
End Sub
```

SL. No	Topic	Covered
Q-24	Select all data in Used Range	

```
Sub Q24()
    Sheet1.UsedRange.Select
End Sub
```

4th Class

1. **Fill Data as per below give Table.**

First Name	Last Name	Salary
Gunjan	Singh	234234
Mohan	Raj	3432
Mona	Sharma	32434
		34324
Raju	Kumar	234234
	Sharma	3434
Sona		34342

```
Private Sub CommandButton1_Click()
Dim Fname, Lname As String
Dim Salary As Long
Dim i, J, K, L As Integer

        Fname = VBA.InputBox("Enter Your First Name")
        Lname = VBA.InputBox("Enter Your Last Name")
        Salary = VBA.InputBox("Enter Your Salary")

                i = Sheet1.Cells(Rows.Count, "A").End(xlUp).Row + 1
                J = Sheet1.Cells(Rows.Count, "B").End(xlUp).Row + 1
                K = Sheet1.Cells(Rows.Count, "C").End(xlUp).Row + 1

        L = Application.WorksheetFunction.Max(i, J, K)

        Sheet1.Cells(L, "A").Value = Fname
        Sheet1.Cells(L, "B").Value = Lname
        Sheet1.Cells(L, "C").Value = Salary

End Sub
```

Fill Data as per below give Table.

Gunjan	Singh	234234
First Name	**Last Name**	**Salary**

```
Private Sub CommandButton1_Click()
Dim Fname, Lname As String
Dim Salary As Long
Dim i, J, K, L As Integer

        Fname = VBA.InputBox("Enter Your First Name")
        Lname = VBA.InputBox("Enter Your Last Name")
        Salary = VBA.InputBox("Enter Your Salary")

                i = Sheet2.Cells(1, 1).End(xlDown).Row - 1
                J = Sheet2.Cells(1, 2).End(xlDown).Row - 1
                K = Sheet2.Cells(1, 3).End(xlDown).Row - 1


        Sheet2.Cells(i, "A").Value = Fname
        Sheet2.Cells(J, "B").Value = Lname
        Sheet2.Cells(K, "C").Value = Salary

End Sub
```

2. **Fill Data as per below give Table.**

Full Name	Last Name	Email	Mob			

```
Private Sub CommandButton1_Click()
Dim Header As String
Dim i As Integer
        Header = VBA.InputBox("Enter Header")
        i = Sheet3.Cells(1, Columns.Count).End(xlToLeft).Column + 1
        Sheet3.Cells(1, i).Value = Header
End Sub
```

4. **Fill Data as per below give Table.**

		Raju	Ranu	Ramu	**Mohani**

```
Private Sub CommandButton1_Click()
Dim Header As String
Dim i As Integer
        Header = VBA.InputBox("Enter Header")
        i = Sheet4.Cells(1, 1).End(xlToRight).Column - 1
        Sheet4.Cells(1, i).Value = Header
End Sub
```

WorkSheet Function

1. **Count**

```
Private Sub CommandButton1_Click()
Sheet1.Range("H6").Value = Application.WorksheetFunction.Count(Sheet1.Range("C2:C21"))
End Sub
```

2. **CountA**

```
Private Sub CommandButton2_Click()
        Sheet1.Range("H7").Value = Application.WorksheetFunction.CountA(Sheet1.Range("A2:C21"))
End Sub
```

3. **Countif**

```
Private Sub CommandButton3_Click()
        Dim rng As Range
        Set rng = Sheet1.Range("A2:A21")
        Sheet1.Range("H8").Value = Application.WorksheetFunction.CountIf(rng, [e2])
End Sub
```

4. **Counifs**

```
Private Sub CommandButton4_Click()
        Dim rng, rng1 As Range
```

```
        Set rng = Sheet1.Range("A2:A21")
        Set rng1 = Sheet1.Range("B2:B21")
        Sheet1.Range("H9").Value = Application.WorksheetFunction.CountIfs(rng, [e2], rng1, [f2])
End Sub
```

5. Sum

```
Private Sub CommandButton5_Click()
        Sheet1.Range("H10").Value = Application.WorksheetFunction.Sum(Sheet1.Range("C2:C21"))
End Sub
```

6. Sumif

```
Private Sub CommandButton6_Click()
        Dim rng, rng1 As Range
        Set rng = Sheet1.Range("A2:A21")
        Set rng1 = Sheet1.Range("C2:C21")
        Sheet1.Range("H11").Value = Application.WorksheetFunction.SumIf(rng, [e2], rng1)
End Sub
```

7. Sumifs

```
Private Sub CommandButton7_Click()
        Dim rng, rng1, rng2 As Range
        Set rng = Sheet1.Range("A2:A21")
        Set rng1 = Sheet1.Range("C2:C21")
        Set rng2 = Sheet1.Range("B2:B21")
        Sheet1.Range("H12").Value = Application.WorksheetFunction.SumIfs(rng1, rng, [e2], rng2, [f2])
End Sub
```

8. Min

```
Private Sub CommandButton8_Click()
        Sheet1.Range("H16").Value = Application.WorksheetFunction.Min(Sheet1.Range("C2:C21"))
End Sub
```

9. Small

```
Private Sub CommandButton9_Click()
        Dim rng1 As Range
```

```
    Set rng1 = Sheet1.Range("C2:C21")
    Sheet1.Range("H18").Value = Application.WorksheetFunction.Small(rng1, 2)
End Sub
```

10. Text

```
Private Sub CommandButton10_Click()
Sheet1.Range("H20").Value = Application.WorksheetFunction.Text(Sheet1.Range("G20"), "MMM")
End Sub
```

Sorting & Filter

SL. No.	Topic
1	Basic Sorting

```
Private Sub CommandButton1_Click()
    Dim rng As Range
    Dim sh As Worksheet
    Set sh = ThisWorkbook.Sheets("Basic Sorting")
    Set rng = sh.Range("A1:C" & sh.Cells(Rows.Count, "C").End(xlUp).Row)
    rng.Sort Key1:=sh.Range("A1"), order1:=xlDescending, Header:=xlYes
End Sub
```

SL. No.	Topic
2	Advance Sorting

```
Private Sub CommandButton2_Click()
    Dim rng As Range
    Dim sh As Worksheet
    Set sh = ThisWorkbook.Sheets("Advance Sorting")
    Set rng = sh.Range("A1:C" & sh.Cells(Rows.Count, "C").End(xlUp).Row)
    rng.Sort Key1:=sh.Range("A1"), order1:=xlDescending, key2:=sh.Range("B1"),
    order2:=xlAscending, Header:=xlYes
End Sub
```

SL. No.	Topic
3	Custom Sorting

```
Private Sub CommandButton2_Click()
```

```
    Dim sh As Worksheet
    Dim rng As Range
    Set sh = ThisWorkbook.Sheets("Custom Sorting")
    Set rng = sh.Range("A2")
    Application.AddCustomList listarray:=sh.Range("E2:E6")
    rng.Sort Key1:=sh.Range("a2"), order1:=xlAscending,
    ordercustom:=Application.CustomListCount + 1, Orientation:=xlSortColumns
End Sub
```

SL. No.	Topic
4	Auto Filter

```
Private Sub CommandButton2_Click()
    Range("A1:C1").AutoFilter
End Sub
```

SL. No.	Topic
5	Advance Filtering

Note-> Option Explicit => General Declaration

```
Private Sub CommandButton2_Click()
    Dim sh As Worksheet
    Dim rng As Range
    Set sh = ThisWorkbook.Sheets("AdvanceFilter2")
    Set rng = sh.Range("A1:C" & sh.Cells(Rows.Count, "C").End(xlUp).Row)
    rng.AdvancedFilter Action:=xlFilterCopy, criteriarange:=sh.Range("H1:J3"),
    copytorange:=sh.Range("M1"), unique:=True
End Sub
```

SL. No.	Topic
6	Remove Duplicates

```
Private Sub CommandButton2_Click()
```

```
    Dim sh As Worksheet
    Dim rng As Range
    Set sh = ThisWorkbook.Sheets("RemoveDuplicates")
    Set rng = sh.Range("A1:A" & sh.Cells(Rows.Count, "A").End(xlUp).Row)
    rng.AdvancedFilter Action:=xlFilterCopy, copytorange:=sh.Range("N1"), unique:=True
End Sub
```

SL. No.	Topic
7	Conditional Formating

```
Private Sub CommandButton2_Click()
    Dim sh As Worksheet
    Dim rng As Range
    Set sh = ThisWorkbook.Sheets("CanditionalFormatting")
    Set rng = sh.Range("a1:A" & sh.Cells(Rows.Count, "A").End(xlUp).Row)
    Dim UV As UniqueValues
    Set UV = rng.FormatConditions.AddUniqueValues
    UV.DupeUnique = xlUnique
    UV.Interior.Color = vbGreen
    UV.DupeUnique = xlDuplicate
    UV.Interior.Color = vbRed
End Sub
```

SL. No.	Topic
8	Series (Fill)

```
Private Sub CommandButton2_Click()
    Dim sh As Worksheet
    Set sh = ThisWorkbook.Sheets("Series")
    sh.Range("C1") = 0
    sh.Range("C1").Select
    Selection.DataSeries rowcol:=xlColumns, Type:=xlLinear, step:=1, stop:=200
End Sub
```

SL. No.	Topic
9	Text to Columns (Fixed with/Delimited)

```
Private Sub CommandButton2_Click()
        Dim sh As Worksheet
        Dim rng As Range
        Set sh = ThisWorkbook.Sheets("TextotColumn")
        Set rng = sh.Range("a2:A" & sh.Cells(Rows.Count, "A").End(xlUp).Row)
        rng.TextToColumns Destination:=sh.Range("C2"), DataType:=xlDelimited, comma:=True,
        other:=True, otherchar:="&"
End Sub
```

SL. No.	Topic
10	Validation

```
Private Sub CommandButton2_Click()
        Range("D10").Select
        Selection.Validation.Delete
        Selection.Validation.Add Type:=xlValidateList, Formula1:="=A1:A5"
End Sub
```

SL. No.	Topic
11	Insert Comment

```
Private Sub CommandButton2_Click()
        ActiveCell.AddComment "Created by VBA Team"
End Sub
```

Event

Private Sub Workbook_Activate()

```
        'Application.DisplayFullScreen = False
        'Application.DisplayFormulaBar = True
        ActiveWindow.DisplayGridlines = True
        ActiveWindow.DisplayHeadings = True
        ActiveWindow.DisplayHorizontalScrollBar = True
        ActiveWindow.DisplayVerticalScrollBar = True
        ActiveWindow.DisplayWorkbookTabs = True
        'ActiveWindow.FreezePanes =
        'ActiveWindow.PrintOut = True
End Sub
```

3. **Double click working.**

```
Private Sub Workbook_SheetBeforeDoubleClick(ByVal Sh As Object, ByVal Target As Range, Cancel As
Boolean)
                If Target.Value <> "" Then
                        Target.Interior.Color = vbRed
                        Else
                        Exit Sub
                End If
End Sub
```

4. **Right Click Disable**

```
Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object, ByVal Target As Range, Cancel As
Boolean)
        If Target.Value <> "" Then
        Target.Interior.Color = vbRed
        Else
                Exit Sub
        End If
End Sub
```

5. **Block Print out.**

```
        Private Sub Workbook_BeforePrint(Cancel As Boolean)
                Cancel = True
                MsgBox ("Print Not Allowed")
```

```
End Sub
```

6. **Block Save as.**

```
Private Sub Workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
    Cancel = False
    MsgBox ("Save Not Allowed")
End Sub
```

Multiple Sheet's Data Copy _ Paste in a Sheet

At first Create Module

```
Private Sub CommandButton1_Click()
        Call QueryCall
        Selection.Copy sh.Cells(i, 1)
        Selection.Interior.Color = vbBlue
End Sub
```

---

1. **Copy data & Pest**

```
Private Sub CommandButton1_Click()
Call QueryCall
Selection.Copy sh.Cells(i, 1)
End Sub
```

2. **Copy data & Pest after color data which has copied**

```
Private Sub CommandButton1_Click()
        Call QueryCall
        Selection.Copy sh.Cells(i, 1)
        Selection.Interior.Color = vbBlue
End Sub
```

3. **Cut Date & Pest**

```
Private Sub CommandButton1_Click()
        Call QueryCall
        Selection.Copy sh.Cells(i, 1)
```

```
        Selection.EntireRow.Delete
End Sub
```

Loop

For Next	Limited
For Each	Object

Do While	Unlimited
Do Until	Unlimited
Do Went	Unlimited

1. **For Next**

a). Eg

```
Sub Loop1()
        Dim R, C As Integer
        For R = 1 To 5
        For C = 1 To 5
        Sheet2.Cells(R, C).Value = "*"
        Next
        Next
End Sub
```

b). Eg

```
Sub Loop2()
        Dim R, C As Integer
        For R = 1 To 5
        For C = 1 To R
        Sheet3.Cells(R, C).Value = "*"
        Next
        Next
End Sub
```

c). Eg

```
Sub Loop3()
Dim R, C As Integer
For R = 1 To 5
For C = 6 - R To 5
Sheet4.Cells(R, C).Value = "*"
Next
 Next
End Sub
```

d).Eg

```
Sub Loop4()
        Dim R, C As Integer
        For R = 1 To 5
```

```
        For C = 1 To 6 - R
        Sheet5.Cells(R, C).Value = "*"
        Next
        Next
End Sub
```

e). Eg

```
Sub Loop5()
        Dim R, C As Integer
        For R = 1 To 5
        For C = R To 5
        Sheet6.Cells(R, C).Value = "*"
        Next
        Next
End Sub
```

f). Eg

```
Sub Loop6()
        Dim R, j, C As Integer
        j = 4
        For R = 0 To j
        For C = 0 To j
        ActiveCell.Offset(R, -C).Value = "*"
        ActiveCell.Offset(R, C).Value = "*"
        Next
        j = j - 1
        Next
End Sub
```

Sr. No	Agenda
1	Value Showing in Msg Box by Loop

```
Sub Loop1()
        Dim i As Integer
        For i = 1 To 5
```

```
        MsgBox i
        Next
End Sub
```

Sr. No	Agenda
2	WorkSheet Value Show in Msg box

```
Sub Loop2()
        Dim i As Integer
        For i = 1 To 10
        MsgBox Sheet1.Cells(i, 2).Value
        Next
End Sub
```

Sr. No	Agenda
3	Sheet Name Show in Msg box

```
Sub Loop3()
        Dim i As Integer
        For i = 1 To Sheets.Count
        MsgBox Sheets(i).Name
        Next
End Sub
```

Sr. No	Agenda
4	Work Sheet ReName By Ref

```
Sub Loop4()
        Dim i As Integer
        Dim sh As Worksheet
        Set sh = ThisWorkbook.Sheets(1)
        For i = 1 To sh.Cells(Rows.Count, "C").End(xlUp).Row
Sheets(i).Name = sh.Cells(i, 3).Value
Next
End Sub
```

Sr. No	Agenda
5	WorkSheet Name Paste in Excel Sheet

```
Sub Loop5()
        Dim i As Integer
        Dim sh As Worksheet
```

```
        Set sh = ThisWorkbook.Sheets(1)
        For i = 1 To Sheets.Count
        sh.Cells(i, 3).Value = Sheets(i).Name
        Next
End Sub
```

Sr. No	Agenda
6	Sheet add and rename by ref

```
Sub Loop6()
        Dim i As Integer
        Dim sh As Worksheet
        Set sh = ThisWorkbook.Sheets(1)
        For i = 1 To sh.Cells(Rows.Count, "D").End(xlUp).Row

        Sheets.Add after:=Sheets(Sheets.Count)
        Sheets(i).Name = sh.Cells(i, "D").Value
        Next
End Sub
```

Sr. No	Agenda
7	WorkSheet Grouping

```
Sub Loop7()
        Dim i As Integer
        For i = 1 To ThisWorkbook.Sheets.Count
        Worksheets.Select (False)
        Next
End Sub
```

Sr. No	Agenda
8	All WorkSheet Select

```
Sub Loop8()
        Dim i As Integer
        For i = 1 To Sheets.Count
```

```
        Sheets(i).Activate
        Cells.Select
        Next
End Sub
```

Sr. No	Agenda
9	Copy Red Color Value only

```
Sub Loop9()
        Dim i, j As Integer
        j = 2
        For i = 1 To Range("A100").End(xlUp).Row
                If Cells(i, "A").Font.Color = vbRed Then
                Cells(j, "B").Value = Cells(i, "A").Value
        j = j + 1
        End If
        Next
End Sub
```

For Each-1

Sr. No	Agenda	Status
1	Remove Naming Range from active sheet	

```
Private Sub CommandButton1_Click()
        Dim N As name
        Dim str As String
        str = "Loop1"
        For Each N In ActiveWorkbook.Names
        If N.RefersToRange.Worksheet.name = str Then
        N.Delete
        End If
        Next
End Sub
```

Sr. No	Agenda	Status
2	Remove All Naming range from Active workbook	

```
Private Sub CommandButton1_Click()
        Dim N As name
        For Each N In ActiveWorkbook.Names
        ActiveWorkbook.Names(N.name).Delete
```

```
        Next
End Sub
```

Sr. No	Agenda	Status
3	Remove Enter	

```
Sub Re_Enter()
Dim rng As Range
For Each rng In Selection
rng.Value = VBA.Trim(rng)
rng.Value = Application.WorksheetFunction.Clean(rng)
Next
End Sub
```

Or

```
Private Sub CommandButton1_Click()
        Dim rng As Range
        For Each rng In Selection
        rng.Value = Application.WorksheetFunction.Clean(rng)
        rng.Value = Application.WorksheetFunction.Trim(rng)
        Next
End Sub
```

Sr. No	Agenda	Status
4	Remove Formulas / Value Pest	

```
Private Sub CommandButton1_Click()
        Dim sh As Worksheet
        On Error Resume Next
        For Each sh In ActiveWorkbook.Sheets
```

```
        sh.Cells.SpecialCells (xlCellTypeFormulas)
        sh.Cells.SpecialCells(xlCellTypeFormulas).Value = sh.Cells.SpecialCells(xlCellTypeFormulas).Value
Next
End Sub
```

Sr. No	Agenda	Status
5	Disable right-click	

```
Sub D_RightC()
        Dim CB As CommandBar
        For Each CB In CommandBars
        If CB.Type = msoBarTypePopup Then CB.Enabled = True
        Next
End Sub
```

Sr. No	Agenda	Status
6	Converter Upper/Lower/Proper Case	

```
Sub case_Conv()
        Dim rng As Range
        For Each rng In Selection
        'rng.Value = VBA.StrConv(rng, vbUpperCase)
        'rng.Value = VBA.StrConv(rng, vbLowerCase)
        rng.Value = VBA.StrConv(rng, vbProperCase)
        Next
End Sub
```

Sr. No	Agenda	Status
7	Reduce all Comment Box Size	

```
Sub ResizeComment()
        Dim rng As Range
        For Each rng In Cells.SpecialCells(xlCellTypeComments)
        rng.Comment.Visible = True
        rng.Comment.Shape.TextFrame.AutoSize = True
```

```
        Next
End Sub
```

Sr. No	Agenda	Status
9	Colored Specific Name in a range	

```
Sub CloseAllExcelSheet()
        Dim xWB As Workbook
        Application.ScreenUpdating = False
        For Each xWB In Application.Workbooks
          If Not (xWB Is Application.ActiveWorkbook) Then
            xWB.Close
          End If
        Next
        Application.ScreenUpdating = True
End Sub
```

Sr. No	Agenda	Status
10	Remove entire row of 0 value	

```
Sub RemoveZeroValue()
        Dim rng As Range
        Dim cel As Range
        Set rng = ActiveSheet.Range("A3:A1000")
        For Each cel In rng
        If cel.Value = 0 Then
        cel.EntireRow.Delete
        End If
        Next
End Sub
```

Sr. No	Agenda	Status
11	Range wise sheet Add & ReName	

```
Sub rangewise_sheetname()
        Dim rng As Range
        Dim cell As Range
        Set rng = ActiveSheet.Range("A2:a4")
        For Each cell In rng
```

```
        Sheets.Add.name = cell.Value
        Next
End Sub
```

Sr. No	Agenda	Status
12	Remove Comments in Active Sheet	

```
Private Sub CommandButton1_Click()
Dim comm As Comment
For Each comm In ActiveSheet.Comments
comm.Delete
Next
End Sub
```

OR

```
Sub RemoveComment()
        Dim cmt As Comment
        For Each cmt In ActiveSheet.Comments
        cmt.Delete
        Next cmt
End Sub
```

For Next Loop

a)

Mix Data	Output
31m312313m13m3m1j123jufhg8130685560fad78af6a	8130685560
dfag7s856jdjhw9000987432rqer7qr87ds8f7adf232323	9000987432
df97f6a86ad5g899g899g9df9g9643206787adf76afd76	9643206787
opi432o8826861950f8adf8adf76fdfuadyuadfuirwenhj	8826861950
qwerjqhrterqjerhqejrhqer8897898778rq3434i2u3424	8897898778

```
Sub GetNum()
        Dim i As Integer, j
        Dim v As Variant
        For i = 2 To Cells(Rows.Count, "A").End(xlUp).Row
        For j = 1 To Len(Range("A" & i))
        If IsNumeric(Mid(Range("A" & i), j, 10)) = True Then
        Range("B" & i) = Mid(Range("A" & i), j, 10)
        GoTo v
        End If
        Next
        v:
        Next
End Sub
```

b)

11	Worksheet Add and rename as per Date
12	Multiple Sheet Delete

Year	Month
2019	Nov

```
Sub AddSheet()
        Dim i As Long
        Dim Sdate, Edate As Date
        Sdate = VBA.DateSerial(Sheet1.Range("A2"), Month(WorksheetFunction.EoMonth(1 &
        Sheet1.Range("B2"), 0)), 1)
        Edate = WorksheetFunction.EoMonth(Sdate, 0)
        For i = Sdate To Edate
        If VBA.Weekday(i, 2) < 6 Then
        Sheets.Add after:=Sheets(Sheets.Count)
        Sheets(Sheets.Count).Name = VBA.Format(i, "dd-mmm-yy")
        End If
        Next
        Sheet1.Activate
End Sub
```

C) Data Paste in Master Sheet to the deference sheet

```
Sub Datacompile()
        Dim sh As Worksheet
```

```
        Dim i, j As Integer
        Set sh = ThisWorkbook.Sheets("Mastersheet")
        sh.Range("A2:m" & sh.Cells(Rows.Count, "M").End(xlUp).Row).Clear
        For i = 2 To Sheets.Count
        j = sh.Cells(Rows.Count, "A").End(xlUp).Row + 1
        Sheets(i).Range("A2:M" & Sheets(i).Cells(Rows.Count, "M").End(xlUp).Row).Copy sh.Cells(j, 1)
        Next
End Sub
```

D) Pest Data according to filter eg by region, by city, by zone. & create sheet name

```
Private Sub CommandButton2_Click()
        Dim sh As Worksheet
        Dim i As Integer
        Dim rng As Range
        Set sh = ThisWorkbook.Sheets("Mastersheet")
        Set rng = sh.Range("N1:N" & sh.Cells(Rows.Count, "N").End(xlUp).Row)
        rng.AdvancedFilter Action:=xlFilterCopy, copytorange:=sh.Range("T1"), unique:=True

        For i = 2 To sh.Cells(Rows.Count, "T").End(xlUp).Row
        Sheets.Add after:=Sheets(Sheets.Count)
        Sheets(Sheets.Count).Name = sh.Cells(i, "T").Value

        sh.Range("A1:N1").AutoFilter field:=14, Criteria1:=sh.Cells(i, "T").Value, visibledropdown:=False
        sh.Range("A1:N" & sh.Cells(Rows.Count,
        "N").End(xlUp).Row).Cells.SpecialCells(xlCellTypeVisible).Copy Sheets(Sheets.Count).Cells(1, 1)
        Sheets(Sheets.Count).Columns.AutoFit
        Next
End Sub
```

e) Pest Data according to filter eg. by region, by city, by zone. & Save in Folder.

```
Private Sub CommandButton2_Click()
Dim i As Integer
Dim sh As Worksheet
Dim wkb, akb As Workbook
Dim path As String, fullpath
Dim rng As Range
        Set sh = ThisWorkbook.Sheets("Sheet1")
        Set akb = ActiveWorkbook
```

```
        On Error Resume Next
        RmDir "C:\Gunjan"
        MkDir "C:\Gunjan\"
        Application.ScreenUpdating = False
        Set rng = sh.Range("N1:N" & sh.Cells(Rows.Count, "N").End(xlUp).Row)
        rng.AdvancedFilter Action:=xlFilterCopy, copytorange:=sh.Range("S1"), unique:=True
        For i = 2 To sh.Cells(Rows.Count, "S").End(xlUp).Row
        Set akb = Workbooks.Add
        sh.Range("a1:N1").AutoFilter field:=14, Criteria1:=sh.Cells(i, "S").Value, visibledropdown:=False
        sh.Range("a1").CurrentRegion.Copy akb.Sheets(1).Cells(1, 1)
        akb.SaveAs "C:\Gunjan\" & sh.Cells(i, "S").Value & ".xlsx"
        akb.Sheets(1).Columns.AutoFit
        akb.Save
        akb.Close
        Next
End Sub
```

DO WHILE/UNTIL

For Next	Limited
For each	Object
Do while/Do Until	Unlimited
While Wend	

Do while

```
Do While i<=10
Output
```

Do Until

```
1   Do until i>10
    output
```

```
i=i+1
Loop
```

Note : it work when condition is true

```
i=i+1
Loop
```

Note : it work when condition is False

---

1. Create series in C Column

```
Private Sub CommandButton1_Click()
        Dim counter As Integer
        counter = 1
        Do While counter < 10
        Cells(counter, "C").Value = counter
        counter = counter + 1
        Loop
End Sub
```

2. Value Multiply in beside column

1 Multiply 10 =	10
2 Multiply 10 =	20
3 Multiply 10 =	30
4 Multiply 10 =	40
5 Multiply 10 =	50
6 Multiply 10 =	60
7 Multiply 10 =	70
8 Multiply 10 =	80

9 Multiply 10 =	90
10 Multiply 10 =	100

```
Private Sub CommandButton1_Click()
        Dim i As Integer
        Dim EN As Integer
        Dim TT As Integer
        EN = 11
        TT = 10
        i = 1
        Do While i < EN
        answer = i * TT
        Cells(i, "B").Value = i & " Multiply " & TT & " = "
        Cells(i, "B").Offset(, 1).Value = answer
        i = i + 1
        Loop
End Sub
```

3. Sum right side of value

Data1	Data2	Total Sum
27	47	74
26	68	94
38	105	143
56	21	77
77	33	110
109	13	122
38	86	124

```
Private Sub CommandButton1_Click()
        ActiveSheet.Range("c2").Select
        Do While Not VBA.IsEmpty(ActiveCell.Offset(0, -1))
        ActiveCell.Offset(0, 0).Value = ActiveCell.Offset(0, -2).Value + ActiveCell.Offset(0, -
        1).Value
        ActiveCell.Offset(1, 0).Select
Loop
End Sub
```

4. Sum below of Numerical Value

Data1	Data2	Data3	Data4	Data5
	92	31	68	70
234	100	41	40	77
2	63	81	69	1
4	29	32		12
	18	38		2
90	90	45		64

100	100	8	18	58
28	64	63	68	87
	71	1	43	18
	2	79	73	95
	8	28	54	100
52	11		42	42
30	43		73	95

```
Private Sub CommandButton1_Click()
Dim i As Integer
Dim rng As Range
Dim rng1 As Range
i = 1
Do While i <= Range("a1").End(xlToRight).Column
Set rng = Range(Cells(2, i), Cells(Rows.Count, i))
Cells(21, i).Value = WorksheetFunction.Sum(rng)
i = i + 1
Loop

End Sub
```

Create Hyperlink as per Worksheet Name in active sheet

Example
Loop_1
Loop_2
Loop_3
Loop_4
Loop_5

```
Private Sub CommandButton2_Click()
```

```
    Dim i As Integer
    i = 1
    Do While i <= Sheets.Count
    Sheets(6).Range("A" & i).Hyperlinks.Add anchor:=Sheets(6).Cells(i, 1), Address:="",
    SubAddress:="'" & Sheets(i).Name & "'" & "!a1"
    Sheets(6).Range("A" & i).Value = Sheets(i).Name
    i = i + 1
    Loop

End Sub
```

For Each – 2

Sr. No	Agenda
1	Column A is Color , Column B status will be True

```
Private Sub CommandButton1_Click()
    Dim rng As Range, cell
    Set rng = ActiveSheet.Range("A2:A20")
    For Each cell In rng
    'If cell.Interior.Color = vbRed Then
    If cell.Interior.ColorIndex >= 1 Then

    cell.Offset(0, 1).Value = "True"
    End If
    Next
End Sub
```

Sr. No	Agenda
2	Remove Space of Selection Field

```
Private Sub CommandButton1_Click()
    Dim rng As Range
    For Each rng In Selection
    rng.Value = Application.WorksheetFunction.Trim(rng.Value)
    Next
End Sub
```

Sr. No	Agenda
3	Remove Enter of Selection Field

```
Private Sub CommandButton1_Click()
```

```
        Dim rng As Range
        For Each rng In Selection
        rng.Value = Application.WorksheetFunction.Substitute(rng.Value, Chr(10), " ")
        Next
End Sub
```

Sr. No	Agenda
4	Remove Shape of Active Sheet

```
Private Sub CommandButton1_Click()
        Dim sh As Worksheet
        Dim sap As Shape
        For Each sh In ActiveWorkbook.Sheets
        For Each sap In sh.Shapes
        sap.Delete
        Next
        Next
End Sub
```

Sr. No	Agenda
5	Unmerge as per range

```
Private Sub CommandButton1_Click()
        Dim sh As Worksheet
        Dim rng As Range
        For Each sh In ActiveWorkbook.Sheets
        Selection.UnMerge
        Next
End Sub
```

For Each – 3

Sr. No	Agenda
1	Delete Duplicate Value as per Range

```
Private Sub CommandButton1_Click()
        Dim rng As Range, cell
        Set rng = ActiveSheet.Range("B3:G23")
        For Each cell In rng
        If WorksheetFunction.CountIf(rng, cell.Value) > 1 Then
        cell.ClearContents
        End If
        Next
End Sub
```

Sr. No	Agenda
2	Highlight Duplicate Value as per Range

```
Private Sub CommandButton2_Click()
        Dim rng As Range, cell
        Set rng = ActiveSheet.Range("A25:F45")
        For Each cell In rng
        If WorksheetFunction.CountIf(rng, cell.Value) = 1 Then
        cell.Interior.Color = vbGreen
        End If
        Next
End Sub
```

Sr. No	Agenda
3	Skip Blank

42	13	72	21	78		
29	98	26	16			
75	97	25	45	45		
38	60	92				

```
Private Sub CommandButton1_Click()
        Dim rng As Range, cell
        Set rng = Range("A1:H24")
        For Each cell In rng
        If cell.Value = "" Then
        If cell.Offset(0, 1).Value = "" Then
        cell.End(xlToRight).Cut cell
        Else
        cell.Offset(0, 1).Cut cell
        End If
        End If
        Next
End Sub
```

Sr. No	Agenda
4	Sum start cell to last Cell in active cell

```
Private Sub CommandButton1_Click()
        RN = ActiveCell.Row
        CN = ActiveCell.Column
        For Each cell In Selection
        RN = cell.Row
        CN = cell.Column
        For c = 1 To CN - 1
        i = i + Cells(RN, c)
        Next
        cell.Value = i
```

```
        i = 0
        Next
End Sub
```

For Each – 4

Compile data in single sheet of different sheets. & Save

```
Sub Backupsheet()
        Dim path As String
        path = Application.ActiveWorkbook.path
        Application.ScreenUpdating = False
        Application.DisplayAlerts = False
        For Each WS In ThisWorkbook.Sheets
        WS.Copy
        Application.ActiveWorkbook.SaveAs Filename:=path & "\" & WS.Name & ".xlsx"
        Application.ActiveWorkbook.Close False
        Next

End Sub
```

Do Until

Sr. No	Agenda
1	Sheet Name Show in Msgbox

```
Private Sub CommandButton1_Click()
        Dim i As Integer
        Dim j As Integer
        i = 1
        j = Sheets.Count
        Do Until i > j
           MsgBox Sheets(i).Name
              i = i + 1
        Loop
End Sub
```

Sr. No	Agenda
2	Sheet Move each other

```
Private Sub CommandButton1_Click()
        Dim i As Integer
        Dim j As Integer
        Dim k As Integer
        i = 1
        k = 1
        j = Sheets.Count
        Do Until k > j
           Sheets(i).Move after:=Sheets(j)
           MsgBox Sheets(j).Name & " has been moved after " & Sheets(i + 1).Name
           k = k + 1
        Loop
End Sub
```

Sr. No	Agenda
3	Last use row count & Entire Row Delete if found Blank cell

```
Private Sub CommandButton1_Click()
        Dim i As Integer
        Dim j As Integer
        Dim rng As Range
        Set rng = Sheet4.Range("A1:a" & Sheet4.Cells(Rows.Count, "a").End(xlUp).Row)
        i = rng.Count
        MsgBox i
        j = 1
        Do Until j > i
          If Sheet4.Cells(j, 1).Value = "" Then
             Sheet4.Cells(j, 1).EntireRow.Delete
          End If
```

```
            j = j + 1
        Loop
End Sub
        Private Sub CommandButton2_Click()
        Dim i, j As Integer
        Dim rng As Range
        Set rng = Sheets(4).Range("A1:A20")
        i = rng.Count
        j = 1
        Do Until j > i
        If Sheets(4).Cells(j, 1).Value = "" Then
        Sheets(4).Cells(j, 1).EntireRow.Delete
        End If
        j = j + 1
Loop End Sub
```

Sr. No	Agenda
4	Add Comment as per 'C' Column Ref.

```
Private Sub CommandButton2_Click()
        Dim i, j As Integer
        Dim str As String
        Dim rng As Range
        Set rng = Sheets(5).Range("C1:C122")
        i = rng.Count
        j = 1
        Do Until j > i
        str = Sheets(5).Cells(j, 3).Value
        Sheets(5).Cells(j, 4).AddComment str
        j = j + 1
        Loop
End Sub
```

Sr. No	Agenda
5	Digit Sum as per A Column ref

Eg. 23 =>5
42 =>6

```
Private Sub CommandButton1_Click()
        Dim i As Integer
        Dim j As Integer
        Dim k As Integer
        Dim l As Integer
        Dim sh As Worksheet
        Set sh = ThisWorkbook.Sheets(6)
        Set rng = sh.Range("A1:A" & sh.Cells(Rows.Count, "A").End(xlUp).Row)
```

```
        k = rng.Count
        j = 1
        MsgBox k
        With sh
        Do Until j > k
              i = VBA.Len(sh.Cells(j, 1))
              l = 1
              Do Until l > i
                .Cells(j, 2).Value = .Cells(j, 2).Value + Mid(.Cells(j, 1), l, 1)
                l = l + 1
              Loop
               j = j + 1
        Loop
        End With

End Sub
```

Sr. No	Agenda
6	Sheet No Count and Create a New Sheet if not exist in the same workbook

```
Private Sub CommandButton1_Click()
        Dim i As Integer
        Dim str As String
        Dim k As Integer
        k = Sheets.Count
        MsgBox "Total sheets in this workbook " & k
        i = 1
        str = Application.InputBox("Provide Sheet Name")
        Do Until i > k
              If str = "" Then
                 MsgBox "Please provide valid sheet name"
                 Exit Sub
              ElseIf UCase(Sheets(i).Name) = UCase(str) Then
                  MsgBox str & " already exist."
                  Exit Sub
              Else
                  MsgBox "Sheet:" & i & " is not " & str & "."
                  i = i + 1
              End If
        Loop
        Sheets.Add after:=Sheets(Sheets.Count)
        Sheets(Sheets.Count).Name = str
End Sub
```

Sr. No	Agenda
7	Total Data Count & Digit Change in reverse order

```
Private Sub CommandButton2_Click()
Dim i, j, k, l As Integer
Dim sh As Worksheet
```

```
Set sh = ThisWorkbook.Sheets(8)
Set rng = sh.Range("A1:A7")
k = Sheets.Count
j = 1
With sh
Do Until j > k
i = VBA.Len(sh.Cells(j, 1))
l = 1
Do Until l > i
.Cells(j, 2).Value = .Cells(j, 2).Value & Mid(.Cells(j, 1), (i + 1) - l, 1)
l = l + 1
Loop
j = j + 1
Loop
End With
End Sub
```

Sr. No	Agenda
8	Sheet Move in reverse order

`

```
Private Sub CommandButton2_Click()
        Dim i, j, k, l As Integer
        i = 1
        k = 1
        j = 1
        l = Sheets.Count
        Do Until k > l
        If k = 1 Then
        Sheets(i).Move after:=Sheets(l)
        k = k + 1
        Else
        Sheets(i).Move after:=Sheets(l - j)
        j = j + 1
        k = k + 1
        End If
        Loop
End Sub
```

Do Until_2

a) Row Insert as per our need

```
Private Sub CommandButton2_Click()
        Dim i As Integer
        i = InputBox("Enter No of Rows")
        If i <= 0 Then
        Exit Sub
        End If
```

```
        Do Until Selection.Value = ""
        ActiveCell.Offset(1, 0).Range("a1:A" & i).Select
        Selection.Insert shift:=xlDown
        ActiveCell.Offset(i, 0).Select
        Loop
End Sub
```

Do Until_3

All Data Copy and paste in new added WS as per our need. (Duplicate sheet)

```
Private Sub CommandButton2_Click()
        Dim i, j As Integer
        Dim sh As Worksheet
        Set sh = ThisWorkbook.Sheets(2)
        i = Application.InputBox("Enter No of New Sheets")
        j = 1
        With sh
        Do Until j > i
        sh.UsedRange.Copy
        Sheets.Add after:=Sheets(Sheets.Count)
        Sheets(Sheets.Count).Cells(1, 1).Select
        ActiveSheet.Paste
        j = j + 1
        Loop
        End With

End Sub
```

Do While -1

All Workbook Compile in active sheet from a folder

```
Sub Datacompile()
        Dim path As String
        Dim fullpath As String
        Dim fl As String
        Dim wkb As Workbook
        Dim i As Integer
```

```
        Dim sh As Worksheet
        Set sh = ActiveSheet
        path = "C:\Users\Nilesh\Desktop\vba-24th Nov\"
        fl = Dir(path & "*.xlsx")
        Do While fl <> ""
        fullpath = path & fl
        Set wkb = Workbooks.Open(fullpath)
        i = sh.Cells(Rows.Count, "A").End(xlUp).Row + 1
        'wkb.Sheets(1).UsedRange.Copy sh.Cells(i, 1)
        wkb.Sheets(1).Range("A2:N" & wkb.Sheets(1).Cells(Rows.Count,
        "N").End(xlUp).Row).Copy sh.Cells(i, 1)
        wkb.Close
        fl = Dir
        Loop
End Sub
```

Do While -2

All Workbook Compile in active sheet from a folder

```
Private Sub CommandButton1_Click()
        Dim wkb As Workbook
        Dim akb As Workbook
        Dim sh As Worksheet
        Dim i As Integer
        Dim pth As String
        Dim fl As String
        Dim full_pth As String

        Set akb = ActiveWorkbook
        Set sh = ThisWorkbook.Sheets("Colleter")

        pth = "C:\Users\Nilesh\Desktop\vba-24th Nov\New folder\"
        fl = Dir(pth & "*.xlsx")
        Do While fl <> ""
        full_pth = pth & fl

        Set wkb = Workbooks.Open(full_pth)
        i = sh.Cells(Rows.Count, "A").End(xlUp).Row + 1
        wkb.Sheets(1).Cells(1, 1).Select

        wkb.Sheets(1).Range(Selection, Selection.End(xlToRight).Address).Copy sh.Cells(1, 1)
        wkb.Sheets(1).Range("a2").CurrentRegion.Select
        Selection.Offset(1, 0).Resize(Selection.Rows.Count, Selection.Columns.Count).Copy
        sh.Cells(i, 1)

        wkb.Close
        sh.Columns.AutoFit
```

```
        fl = Dir
Loop
```

End Sub

User Define Function

Sr. No	Agenda
1	Count

```
Function Count1(rng As Range)
        Dim cell As Range
        Dim i As Integer
        For Each cell In rng
        If cell.Value <> "" And VBA.IsNumeric(cell.Value) Then
        i = i + 1
        End If
        Next
        Count1 = i
End Function
```

Sr. No	Agenda
2	Countif

```
Function Countif1(rng As Range, abc As Range)
        Dim cell As Range
        Dim i As Integer
        i = 0
        For Each cell In rng
        If VBA.UCase(cell.Value) = VBA.UCase(abc.Value) Then
        i = i + 1
        End If
        Next
        Countif1 = i
End Function
```

Sr. No	Agenda
3	Text Split

```
Function Text1(rng As Range)
        Dim i As Integer
        For i = 1 To Len(rng)
        If VBA.Mid(rng, i, 1) Like "[a-z,A-Z]" Then
        Text1 = Text1 & Mid(rng, i, 1)
        End If
        Next
End Function
```

Sr. No	Agenda
4	Number Split

```
Function Nume(rng As Range)
        Dim i As Integer
        For i = 1 To Len(rng)
        If VBA.Mid(rng, i, 1) Like "[0-9]" Then
        Nume = Nume & VBA.Mid(rng, i, 1)
        End If
        Next
End Function
```

Sr. No	Agenda
5	Special characters Split

```
Function SPL(rng As Range)
        Dim i As Integer
        For i = 1 To Len(rng)
        If Not VBA.Mid(rng, i, 1) Like "[a-z,A-Z,0-9]" Then
        SPL = SPL & Mid(rng, i, 1)
        End If
        Next
End Function
```

Sr. No	Agenda
6	Color count

```
Function Color_count(rng As Range, ctr As Range)
        Dim cell As Range
        Dim i As Integer
        For Each cell In rng
        If cell.Interior.Color = ctr.Interior.Color Then
        i = i + 1
        End If
        Next cell
```

```
    Color_count = i
End Function
```

Sr. No	Agenda
7	Reverse Order

```
Function Retext(rng As Range)
    Retext = VBA.StrReverse(rng)
End Function
```

Sr. No	Agenda
8	Total Day (Apart From Sunday)

Start Date	End Date	Total Day (Apart From Sunday)
11/29/2013	12/29/2013	26

```
Function daycount(sdate As Date, edate As Date)
    Dim i As Long
    Dim j As Integer
    For i = sdate To edate
    If VBA.Weekday(i, vbMonday) <= 6 Then
    j = j + 1
    End If
    Next
    daycount = j
End Function
```

9. Data	Evaluator
2+2-2/2*2	2
35/143-20	-19.75524476
23+10	33
345-12	333
345-110	235
12*23	276

```
Function Eval(rng As Range)
    Eval = Application.Evaluate("=" & rng)

End Function
```

11. Work Book Name

```
Function WKN()
Application.Volatile
WKN = ActiveWorkbook.Name
```

12. Work Sheet Name

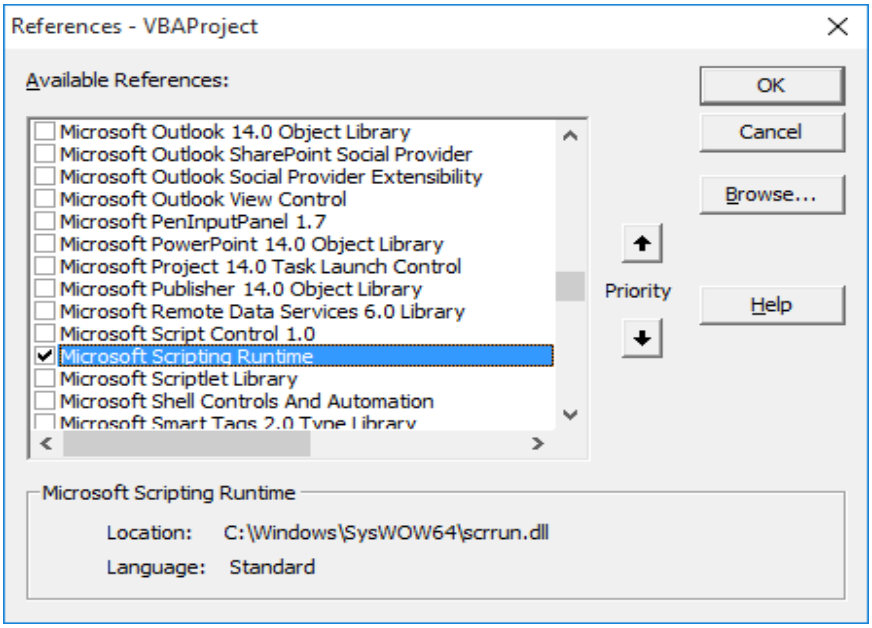
```
Function WSN()
Application.Volatile
WSN = ActiveSheet.Name

End Function
```

Sr. No	Agenda
1	Date Format Change (UDF)

```
Function Day_F(rng As Range)
    Select Case Day(rng.Value)
    Case 1, 21, 31
    Day_F = VBA.Day(rng) & "st" & VBA.Format(rng, " mmm' yy")
    Case 2, 22
    Day_F = VBA.Day(rng) & "nd" & VBA.Format(rng, " mmm' yy")
    Case 3, 23
    Day_F = VBA.Day(rng) & "rd" & VBA.Format(rng, " mmm' yy")
    Case Else
    Day_F = VBA.Day(rng) & "th" & VBA.Format(rng, " mmm' yy")
    End Select
End Function
```

FSO (File system object)

Add this library




```
Sub File_Tracker()
        Dim fso As FileSystemObject
        Dim fol As Folder
        Dim fil As File
        Dim sh As Worksheet
        Dim path As String
        Application.FileDialog(msoFileDialogFolderPicker).Show
        path = Application.FileDialog(msoFileDialogFolderPicker).SelectedItems(1) & "\"
        Set sh = ThisWorkbook.Sheets(2)
        'sh.UsedRange.Clear
```

```
        Set fso = New FileSystemObject
        Set fol = fso.GetFolder(path)
        With sh
        i = 2
        For Each fil In fol.Files
        .Cells(i, 1).Value = fil.Name
        .Cells(i, 2).Value = fil.DateCreated
        .Cells(i, 3).Value = fil.DateLastModified
        .Cells(i, 4).Value = fil.Type
        .Cells(i, 5).Value = fil.Size
        i = i + 1
        Next
        End With
End Sub
```

Note we have a folder name is Test data in that 4 file name is

1. Ram
2. HDD
3. Server
4. Processor

```
Private Sub CommandButton1_Click()
Dim fso As New FileSystemObject
Dim fldr As Folder
Dim fl As File
Dim i As Long
Dim wkb As Workbook
Dim akb As Workbook
Dim sh As Worksheet
Dim sh1 As Worksheet
Set akb = ThisWorkbook

On Error Resume Next

Set fldr = fso.GetFolder("F:\Nilesh\Excel _ Macro Training_Class\1_Training\2_Macro Vs
VBA\VBA-Training\16-File System Object (FSO)\FSO\Test_Data\")
For Each fl In fldr.Files
If fl.Name <> "Mastersheet.xlsx" Then
Workbooks.Open Filename:=fl
Set wkb = ActiveWorkbook
End If

For Each sh In wkb.Sheets
For Each sh1 In akb.Sheets
If sh.Name = sh1.Name Then
i = sh1.Cells(Rows.Count, 1).End(xlUp).Row + 1
sh.Range("A2:F" & sh.Cells(Rows.Count, "F").End(xlUp).Row).Copy sh1.Cells(i, 1)
```

```
End If
Next sh1
wkb.Close
Next sh
Next fl

End Sub
```

Array

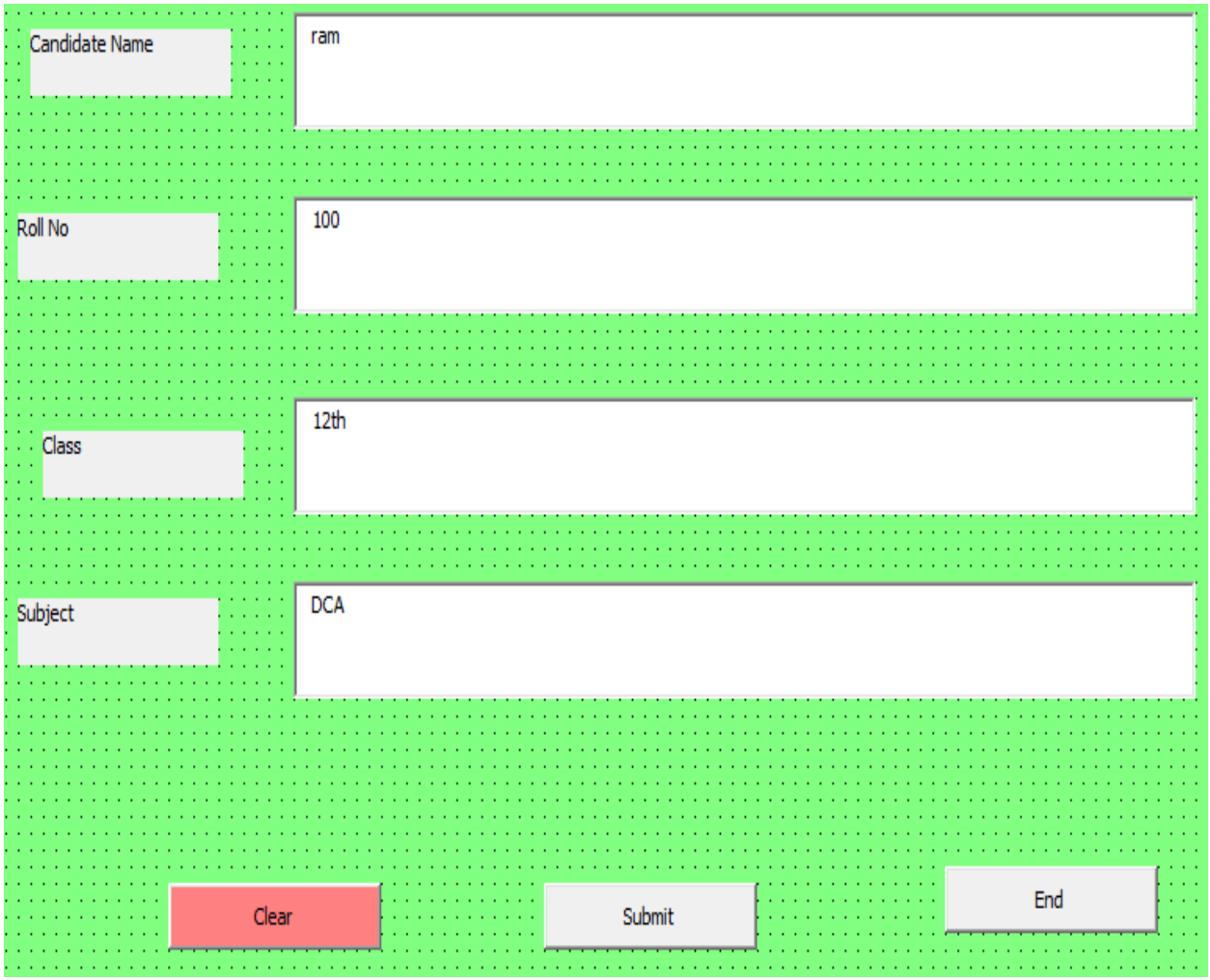
```
Sub Array1()
    Dim arr(6) As String
    Dim i As Integer
    arr(0) = "Sun"
    arr(1) = "Mon"
    arr(2) = "Tue"
    arr(3) = "Wed"
    arr(4) = "Thu"
    arr(5) = "Fri"
    arr(6) = "Sat"
    For i = 0 To UBound(arr())
    MsgBox arr(i)
    Next
End Sub

Sub Arr2()
    Dim sh As Worksheet
    Dim rng
    Set sh = ThisWorkbook.Sheets(2)
    rng = sh.Range("A2:B" & sh.Cells(Rows.Count, "B").End(xlUp).Row)
    For i = LBound(rng) To UBound(rng)
    MsgBox rng(i, 1)
    MsgBox rng(i, 2)
    Next

End Sub

Sub arr3()
    Dim str As String
    Dim arr
    str = "I am VBA User"
    arr = VBA.Split(str, " ")
```

```
    For i = LBound(arr) To UBound(arr)
    MsgBox (arr(i))
    Next
End Sub
```

User Form

Candidate Name	Roll No	Class	Subject
Gunjan	101	Graduation	Excel

VBA (VISUAL BASIC APPLICATION)



Note: For Command Button

```
Private Sub CommandButton1_Click()
    UserForm1.Show
    End Sub
    Form
```

Note :- For Text Box

```
Private Sub CommandButton1_Click()
        TextBox1.Text = ""
```

```
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
End Sub
```

Note:- For Submit Button:-

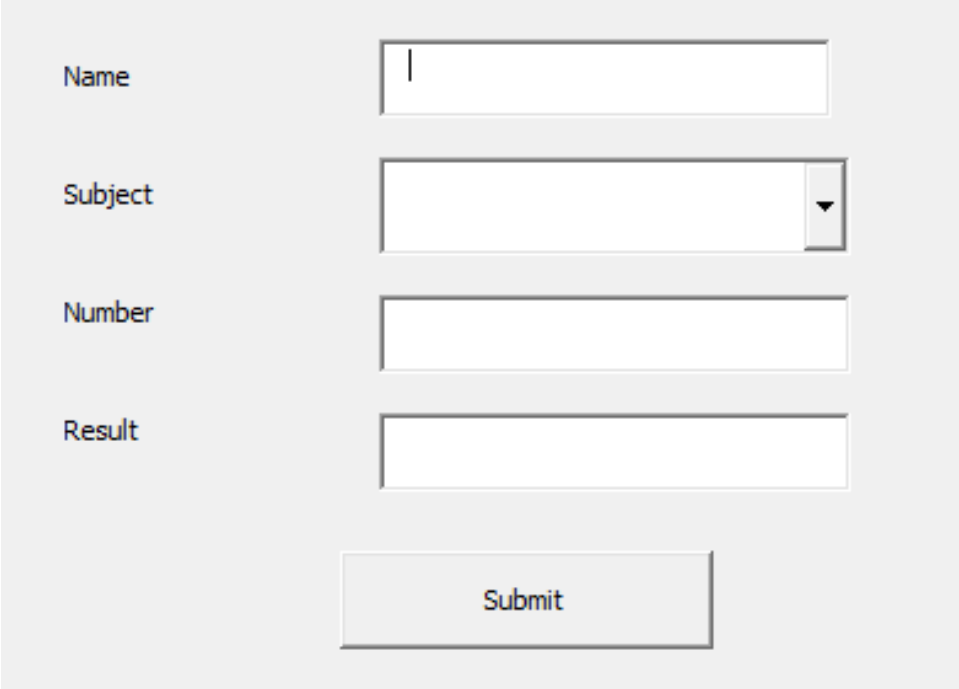
```
Private Sub CommandButton2_Click()
        Dim i As Integer
        i = Sheet1.Cells(Rows.Count, 1).End(xlUp).Row + 1
        Cells(i, 1) = TextBox1.Text
        Cells(i, 2) = TextBox2.Text
        Cells(i, 3) = TextBox3.Text
        Cells(i, 4) = TextBox4.Text

End Sub
```

Note:- For End Button

```
Private Sub CommandButton3_Click()
        Unload Me
End Sub
```

Name	Subject	Number	Result
Raj	English	33	Pass
Raj	English	20	Fail

VBA (VISUAL BASIC APPLICATION)




```
Sub Button1_Click()
UserForm1.Show
End Sub


Private Sub UserForm_Initialize()
ComboBox1.List = Array("English", "Hindi", "Math")

End Sub
```

```
Private Sub CommandButton1_Click()
Dim i As Long
Dim sh As Worksheet
Set sh = ThisWorkbook.Sheets(1)

i = sh.Range("A" & Rows.Count).End(xlUp).Row
```

```
sh.Cells(i + 1, 1).Value = TextBox1.Text
sh.Cells(i + 1, 2).Value = ComboBox1.Value
sh.Cells(i + 1, 3).Value = TextBox2.Text

If TextBox2.Value > 30 = True Then
TextBox3.Text = "Pass"
sh.Cells(i + 1, 4).Value = TextBox3.Text
Else
TextBox3.Text = "Fail"
sh.Cells(i + 1, 4).Value = TextBox3.Text
End If

End Sub
```

IF()

```
Sub abcd()
    Dim math As Long, Sin As Long
    math = Application.InputBox("Enter Marks of Math", "Pasing Criteria 50")
    Sin = Application.InputBox("Enter Marks of Sin", "Pasing Criteria 30")
    If math >= 50 And Sin >= 30 Then
            MsgBox "Pass in both Subject"
                ElseIf Sin >= 30 And math < 50 Then
                    MsgBox "Pass in sin but fail in math"
                ElseIf Sin < 30 And math >= 50 Then
            MsgBox "Fail in sin but Pass in math"

            Else
    MsgBox "Fail in both Subject"
    End If

End Sub
```

Outlook

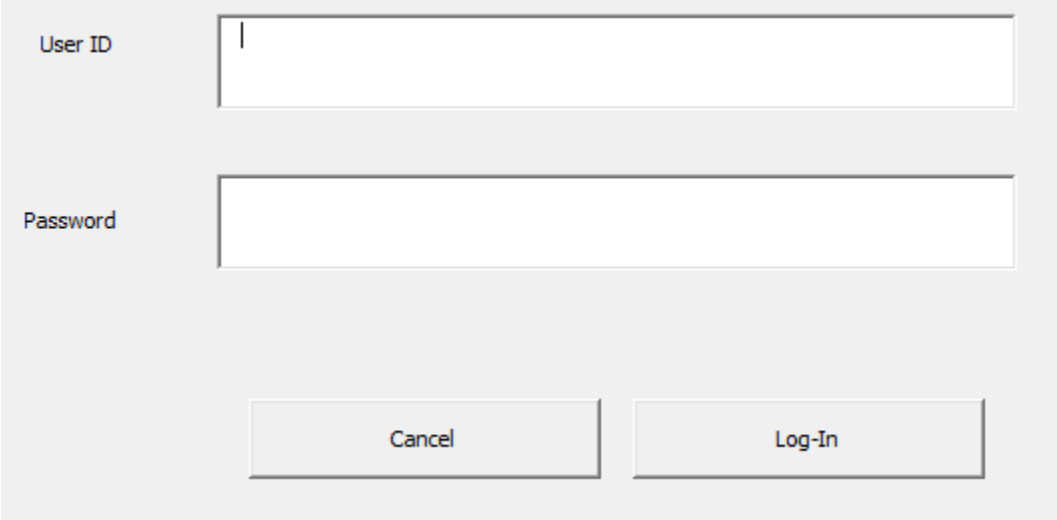
Candidate ID	CandidateName	EnrollmentNumber	NameOfFatherOrHusband	Email ID

6947270	Gandhimadhi Gopi	BWSSC/NKL/B1/008	Gopi	gunjan@gmail.com
6947271	Gowri Veerasamy	BWSSC/NKL/B1/009	Veerasamy	gunjan@gmail.com
6947272	Jayamani Ashogan	BWSSC/NKL/B1/010	Ashogan	gunjan@gmail.com
6947273	Kalaiyarasi Kandhasamy	BWSSC/NKL/B1/011	Kandhasamy	gunjan@gmail.com
6947274	Kavitha Velusamy	BWSSC/NKL/B1/012	Velusamy	gunjan@gmail.com
6947396	Kamatchi B	BWSSC/MANACHANALLUR/B1/07	BALU	Rajesh@yahoo.com
6947397	Kavitha P	BWSSC/MANACHANALLUR/B1/08	PERIYASAMY	Rajesh@yahoo.com
6947398	Kayathri T	BWSSC/MANACHANALLUR/B1/09	THIRUVENGADAM	Rajesh@yahoo.com
6947399	Kirija A	BWSSC/MANACHANALLUR/B1/10	ANNADURAI	Rajesh@yahoo.com
6954011	Prameswari Mahalngam	BWSSC/NKL/B2/023	Mahalngam	Rish@hotmail.com
6954012	Priya Elavarasan	BWSSC/NKL/B2/024	Elavarasan	Rish@hotmail.com
6954013	Rukshana Mansoor Rahman	BWSSC/NKL/B2/025	Mansoor Rahnan	Rish@hotmail.com
6954014	Sasikala Ramasamy	BWSSC/NKL/B2/026	Ramasamy	Rish@hotmail.com
6954015	Sindhuja Murugesan	BWSSC/NKL/B2/027	Murugesan	Rish@hotmail.com
6966478	Sangeetha N	BWSSC/NKL/B3/017	Nallaiyan	Plogix.xl@gmail.com
6966479	Saranya Saravanan	BWSSC/NKL/B3/018	Saravanan	Plogix.xl@gmail.com
6966480	Saranya Subramaniyam	BWSSC/NKL/B3/019	Subramaniyam	Plogix.xl@gmail.com

```
Private Sub CommandButton1_Click()
    Dim sh As Worksheet
    Dim rng As Range
    Dim i As Integer
    Dim path As String
    Dim wkb, tkb As Workbook
```

```
    Dim olapp As Outlook.Application
    Dim mailitem As Outlook.mailitem

    Set tkb = ThisWorkbook
    Set sh = ThisWorkbook.Sheets(1)
    On Error Resume Next
    RmDir "C:\OutlookAuto"
    MkDir "C:\OutlookAuto\"
    Application.ScreenUpdating = False

    Set rng = sh.Range("N1:N" & sh.Cells(Rows.Count, "N").End(xlUp).Row)
    rng.AdvancedFilter Action:=xlFilterCopy, copytorange:=sh.Range("T1"), unique:=True
    Set olapp = New Outlook.Application

    For i = 2 To sh.Cells(Rows.Count, "T").End(xlUp).Row
    Set mailitem = olapp.CreateItem(olMailItem)
    Set wkb = Workbooks.Add

    sh.Range("A1:N1").AutoFilter field:=14, Criteria1:=sh.Cells(i, "T"),
    visibledropdown:=False
    sh.Range("A1:n" & sh.Cells(Rows.Count,
    "N").End(xlUp).Row).SpecialCells(xlCellTypeVisible).Copy wkb.Sheets(1).Cells(1, 1)
    wkb.SaveAs "C:\OutlookAuto\" & sh.Cells(i, "T").Value & ".xlsx"
    wkb.Close
    With mailitem
    .To = sh.Cells(i, "T").Value
    .CC = "abc@gmail.com"
    .BCC = "xyx@yahoo.com"
    .Subject = sh.Range("W1").Value
    .Body = "Dear Sir/Ma'am," & Chr(10) & "Please find the attached Daily MIS Report" &
    Chr(10) & Chr(10) & "Thanks & Regards, " & Chr(10) & "VBA Team"
    .Attachments.Add "C:\OutlookAuto\" & sh.Cells(i, "T").Value & ".xlsx"
    .Display
    .Send
     End With

     Next
End Sub
```

Login ID



For Cancel

```
Private Sub CommandButton1_Click()
    Unload Me
    ActiveWorkbook.Close
    Application.Quit
End Sub
```

For Log-In

```
Private Sub CommandButton2_Click()
    Dim sh As Worksheet
    Set sh = ThisWorkbook.Sheets(1)

    If TextBox1 = sh.Range("A1") Then
    If TextBox2 = sh.Range("b1") Then
    MsgBox "Welcome to you"
    Application.Visible = True
    Unload Me
    Else
    MsgBox "You have Entered Wrong Information, Please try again"
    End If
    End If

End Sub
```

ID, Password will be cell No. A1, b1 which on code