Asset Allocation & Return Forecasting Model

using Excel VBA

```
Option Explicit

' Declare global variables to store data across the entire workbook
Dim currentCapital As Double        ' Store the current capital for each year
Dim numYears As Integer             ' Store the number of years for investment
Dim TotalExpectedAllClasses As Double ' Store the total expected return of all classes
Dim TotalBestAllClasses As Double     ' Store the total best return of all classes
Dim TotalWorstAllClasses As Double    ' Store the total worst return of all classes
Dim resultsString As String          ' String to store results for each year

' Main subroutine for calculating returns for each year
Sub ReturnClassByYears()

   ' Reference to the worksheet where data will be stored
   Dim ws As Worksheet
   Set ws = ThisWorkbook.Sheets("Annual Investment Record")

   ' Find the last used row in column A of the worksheet
   Dim lastRow As Long
   lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

   ' Set the starting row for new data
   Dim startRow As Long
   startRow = lastRow + 1

   ' Prompt the user for the number of years
   numYears = InputBox("Enter the number of years that you want to invest:", "Invest Years", 1)

   ' Process each year
   Dim i As Integer
```

```
    For i = 1 To numYears
        ' Reset totals for each year
        TotalExpectedAllClasses = 0
        TotalBestAllClasses = 0
        TotalWorstAllClasses = 0

        ' Prompt the user for the capital for the current year
        currentCapital = InputBox("Enter the Capital (in Billions) that you want to invest for Year " & i & ":", "Capital to invest for Year " & i)

        ' Store the year and capital in the worksheet after the existing data
        ws.Cells(startRow + i, 1).Value = i             ' Column A is for the year
        ws.Cells(startRow + i, 2).Value = currentCapital  ' Column B is for capital

        ' Reset percentages for each class at the beginning of each year
        Dim class1Pct As Double
        Dim class2Pct As Double
        Dim class3Pct As Double
        Dim class4Pct As Double

    ' Reset percentages for each class at the beginning of each year
        class1Pct = 0
        class2Pct = 0
        class3Pct = 0
        class4Pct = 0

        ' Class 1
        Do
            ' Prompt the user for the percentage of capital allocated for Class 1
            class1Pct = InputBox("Enter the % of Capital allocate for Class 1 for Year " & i & " Remaining: " & FormatNumber(100- class2Pct- class3Pct- class4Pct, 2) & "%", "Class 1 Allocate")

            ' Check if the total percentage exceeds 100%, and prompt for re-entry if necessary
            If class1Pct + class2Pct + class3Pct + class4Pct > 100 Then
                MsgBox "Total percentage exceeds 100%. Please re-enter.", vbExclamation
            End If
        Loop While class1Pct + class2Pct + class3Pct + class4Pct > 100

        ' Store details for Class 1 in a separate worksheet
        ThisWorkbook.Sheets("Class 1 Details").Cells(6, 1).Value = numYears     ' Column A is for the year
        ThisWorkbook.Sheets("Class 1 Details").Cells(6, 2).Value = currentCapital ' Column B is for capital
        ThisWorkbook.Sheets("Class 1 Details").Cells(6, 3).Value = class1Pct      ' Column C is for class1Pct
        ' Calculate returns for Class 1 and its subclasses
        Class1Return class1Pct

        ' Reset percentages for class 2 at the beginning of each year
        class2Pct = 0

        ' Class 2
        Do
            ' Prompt the user for the percentage of capital allocated for Class 2
            class2Pct = InputBox("Enter the % of Capital allocate for Class 2 for Year " & i & "
Remaining: " & FormatNumber(100- class1Pct- class3Pct- class4Pct, 2) & "%", "Class 2
Allocate")

            ' Check if the total percentage exceeds 100%, and prompt for re-entry if necessary
            If class1Pct + class2Pct + class3Pct + class4Pct > 100 Then
                MsgBox "Total percentage exceeds 100%. Please re-enter.", vbExclamation
            End If
        Loop While class1Pct + class2Pct + class3Pct + class4Pct > 100

        ' Store details for Class 2 in a separate worksheet
        ThisWorkbook.Sheets("Class 2 Details").Cells(6, 1).Value = numYears     ' Column A is for
the year
        ThisWorkbook.Sheets("Class 2 Details").Cells(6, 2).Value = currentCapital ' Column B is for
capital
        ThisWorkbook.Sheets("Class 2 Details").Cells(6, 3).Value = class2Pct      ' Column C is for
class2Pct

        ' Calculate returns for Class 2 and its subclasses
        Class2Return class2Pct

        ' Reset percentages for class 3 at the beginning of each year
        class3Pct = 0

        ' Class 3
        Do
            ' Prompt the user for the percentage of capital allocated for Class 3
            class3Pct = InputBox("Enter the % of Capital allocate for Class 3 for Year " & i & "
Remaining: " & FormatNumber(100- class1Pct- class2Pct- class4Pct, 2) & "%", "Class 3
Allocate")

            ' Check if the total percentage exceeds 100%, and prompt for re-entry if necessary
            If class1Pct + class2Pct + class3Pct + class4Pct > 100 Then
                MsgBox "Total percentage exceeds 100%. Please re-enter.", vbExclamation
            End If
        Loop While class1Pct + class2Pct + class3Pct + class4Pct > 100

        ' Store details for Class 3 in a separate worksheet
        ThisWorkbook.Sheets("Class 3 Details").Cells(6, 1).Value = numYears     ' Column A is for
the year
        ThisWorkbook.Sheets("Class 3 Details").Cells(6, 2).Value = currentCapital ' Column B is for
capital
        ThisWorkbook.Sheets("Class 3 Details").Cells(6, 3).Value = class3Pct      ' Column C is for
class3Pct

        ' Calculate returns for Class 3 and its subclasses
        Class3Return class3Pct

        ' Reset percentages for class 4 at the beginning of each year
        class4Pct = 0

        ' Class 4
        Do
            ' Prompt the user for the percentage of capital allocated for Class 4


 class4Pct = InputBox("Enter the % of Capital allocate for FD of Banks for Year " & i & "
Remaining: " & FormatNumber(100- class1Pct- class2Pct- class3Pct, 2) & "%", "FD Banks
Allocate")

            ' Check if the total percentage exceeds 100%, and prompt for re-entry if necessary
            If class1Pct + class2Pct + class3Pct + class4Pct > 100 Then
                MsgBox "Total percentage exceeds 100%. Please re-enter.", vbExclamation
            End If
        Loop While class1Pct + class2Pct + class3Pct + class4Pct > 100

        ' Store details for Class 4 in a separate worksheet
        ThisWorkbook.Sheets("Class 4 Details").Cells(6, 1).Value = numYears     ' Column A is for
the year
        ThisWorkbook.Sheets("Class 4 Details").Cells(6, 2).Value = currentCapital ' Column B is for
capital
        ThisWorkbook.Sheets("Class 4 Details").Cells(6, 3).Value = class3Pct      ' Column C is for
class4Pct

        ' Calculate returns for Class 4
        Class4Return class4Pct

        ' Store the results for the current year
        Dim YearlyTotalExpected As Double
        Dim YearlyTotalBest As Double
        Dim YearlyTotalWorst As Double

        ' Display results for the current year
        YearlyTotalExpected = TotalExpectedAllClasses
        YearlyTotalBest = TotalBestAllClasses
        YearlyTotalWorst = TotalWorstAllClasses

        ' Store the results in the worksheet
        ws.Cells(startRow + i, 3).Value = YearlyTotalExpected ' Column C is for YearlyTotalExpected
        ws.Cells(startRow + i, 4).Value = YearlyTotalBest      ' Column D is for YearlyTotalBest
        ws.Cells(startRow + i, 5).Value = YearlyTotalWorst     ' Column E is for YearlyTotalWorst

        ' Concatenate results for the current year
        resultsString = resultsString & "RETURN FOR Year " & i & "->" & vbCrLf & _
            "  Expected: " & IIf(YearlyTotalExpected < 0, "(" &
FormatNumber(Abs(YearlyTotalExpected), 3) & ")", FormatNumber(YearlyTotalExpected, 3)) &
" billion." & vbCrLf & _
            "  Best: " & IIf(YearlyTotalBest < 0, "(" & FormatNumber(Abs(YearlyTotalBest), 3) & ")",
FormatNumber(YearlyTotalBest, 3)) & " billion." & vbCrLf & _
            "  Worst: " & IIf(YearlyTotalWorst < 0, "(" & FormatNumber(Abs(YearlyTotalWorst), 3) &
")", FormatNumber(YearlyTotalWorst, 3)) & " billion." & vbCrLf & vbCrLf

    Next i

    ' Display the results for all years
    MsgBox resultsString, vbInformation

End Sub


Sub Class1Return(class1Pct As Double)
    ' Declare variables
    Dim capital As Double
    Dim year As Double
    Dim SubPcts(1 To 3) As Double
    Dim SubMinRates(1 To 3) As Double
    Dim SubMaxRates(1 To 3) As Double
    Dim SubNames(1 To 3) As String
    Dim resultMessage As String
    Dim Class1TotalExpected As Double
    Dim Class1TotalBest As Double
    Dim Class1TotalWorst As Double

    ' Input total capital and year
    capital = currentCapital
    year = numYears

    ' Set reference to the "Class 1 Details" worksheet
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Class 1 Details")

    ' Find the last used row in column A of "Class 1 Details"
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    ' Increment lastRow by 1 to get the startRow
    Dim startRow As Long
    startRow = lastRow + 1

    ' Store the year, capital, and class1Pct in the "Class 1 Details" worksheet
    ws.Cells(startRow, 1).Value = year
    ws.Cells(startRow, 2).Value = capital
    ws.Cells(startRow, 3).Value = class1Pct

    ' Set Sub allocations and rates
    SubNames(1) = "AIandIT"
    SubMinRates(1) = 0.08
    SubMaxRates(1) = 0.15

    SubNames(2) = "CONSUMER"
    SubMinRates(2) =-0.05
    SubMaxRates(2) = 0.03

    SubNames(3) = "ENERGY"
    SubMinRates(3) = 0
    SubMaxRates(3) = 0.08

    ' Prompt the user for SubPcts for Class 1
    Dim i As Integer
    For i = 1 To 3
        Do
            SubPcts(i) = InputBox("Enter the % for " & SubNames(i) & vbCrLf & "Remaining: " &
FormatNumber(100- SubPcts(1)- SubPcts(2)- SubPcts(3), 2) & "%", "Class 1")
            If SubPcts(1) + SubPcts(2) + SubPcts(3) > 100 Then
                MsgBox "Total percentage exceeds 100%. Please re-enter.", vbExclamation
            End If
        Loop While SubPcts(1) + SubPcts(2) + SubPcts(3) > 100

        ' Record SubPcts in cells D, H, L (adjusted for the starting column)
        ws.Cells(startRow, 4 + (i - 1) * 4).Value = SubPcts(i)
    Next i

    ' Loop through Subs to calculate returns
    For i = 1 To 3
        Dim Worst As Double
        Dim Best As Double
        Dim Expected As Double

        ' Calculate worst, best, and expected returns for each subclass
        Worst = capital * (class1Pct / 100) * (SubPcts(i) / 100) * SubMinRates(i)
        Best = capital * (class1Pct / 100) * (SubPcts(i) / 100) * SubMaxRates(i)
        Expected = (Worst + Best) / 2

        ' Add Sub values to Class 1 totals
        Class1TotalExpected = Class1TotalExpected + Expected
        Class1TotalBest = Class1TotalBest + Best
        Class1TotalWorst = Class1TotalWorst + Worst

        ' Construct the result message for each Sub
        resultMessage = resultMessage & "  " & SubNames(i) & ":" & vbCrLf & _
                        "    Expected: " & FormatNumber(Expected, 3) & " billion." & vbCrLf & _
                        "    Best: " & FormatNumber(Best, 3) & " billion." & vbCrLf & _
                        "    Worst: " & IIf(Worst < 0, "(" & FormatNumber(Abs(Worst), 3) & ")",
FormatNumber(Worst, 3)) & " billion." & vbCrLf

        ' Record the expected value in cells E, I, M (adjusted for the starting column)
        ws.Cells(startRow, 5 + (i - 1) * 4).Value = Expected
        ws.Cells(startRow, 6 + (i - 1) * 4).Value = Best
        ws.Cells(startRow, 7 + (i - 1) * 4).Value = Worst
    Next i

    ' Construct the result message for Class 1
    resultMessage = "CLASS 1 RETURN CASE FOR YEAR " & year & "->" & vbCrLf & _
                    "  Expected: " & FormatNumber(Class1TotalExpected, 3) & " billion." & vbCrLf & _
                    "  Best: " & FormatNumber(Class1TotalBest, 3) & " billion." & vbCrLf & _
                    "  Worst: " & FormatNumber(Class1TotalWorst, 3) & " billion." & vbCrLf & _
                    "Sub Class->" & vbCrLf & resultMessage

    ' Update overall totals for all classes
    TotalExpectedAllClasses = TotalExpectedAllClasses + Class1TotalExpected
    TotalBestAllClasses = TotalBestAllClasses + Class1TotalBest
    TotalWorstAllClasses = TotalWorstAllClasses + Class1TotalWorst

    ' Display the result in one message box
    MsgBox resultMessage, vbInformation
End Sub


Sub Class2Return(class2Pct As Double)
    ' Declare variables
    Dim capital As Double
    Dim year As Double
    Dim SubPcts(1 To 3) As Double
    Dim SubMinRates(1 To 3) As Double
    Dim SubMaxRates(1 To 3) As Double
    Dim SubNames(1 To 3) As String
    Dim resultMessage As String
    Dim Class2TotalExpected As Double
    Dim Class2TotalBest As Double
    Dim Class2TotalWorst As Double

    ' Input total capital and year
    capital = currentCapital
    year = numYears

    ' Set reference to the "Class 2 Details" worksheet
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Class 2 Details")

    ' Find the last used row in column A of "Class 2 Details"
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    ' Increment lastRow by 1 to get the startRow
    Dim startRow As Long
    startRow = lastRow + 1

    ' Store the year, capital, and class2Pct in the "Class 2 Details" worksheet
    ws.Cells(startRow, 1).Value = year
    ws.Cells(startRow, 2).Value = capital
    ws.Cells(startRow, 3).Value = class2Pct

    ' Set Sub allocations and rates for Class 2
    SubNames(1) = "BANKS"
    SubMinRates(1) =-0.01
    SubMaxRates(1) = 0.08

    SubNames(2) = "ENERGY"
    SubMinRates(2) = 0
    SubMaxRates(2) = 0.08

    SubNames(3) = "AI"
    SubMinRates(3) = 0.08
    SubMaxRates(3) = 0.15

    ' Prompt the user for SubPcts for Class 2
    Dim i As Integer
    For i = 1 To 3
        Do
            SubPcts(i) = InputBox("Enter the % for " & SubNames(i) & vbCrLf & "Remaining: " &
FormatNumber(100- SubPcts(1)- SubPcts(2)- SubPcts(3), 2) & "%", "Class 2")
            If SubPcts(1) + SubPcts(2) + SubPcts(3) > 100 Then
                MsgBox "Total percentage exceeds 100%. Please re-enter.", vbExclamation
            End If
        Loop While SubPcts(1) + SubPcts(2) + SubPcts(3) > 100

        ' Record SubPcts in cells D, H, L (adjusted for the starting column)
        ws.Cells(startRow, 4 + (i - 1) * 4).Value = SubPcts(i)
    Next i

    ' Loop through Subs to calculate returns
    For i = 1 To 3
        Dim Worst As Double
        Dim Best As Double
        Dim Expected As Double

        ' Calculate worst, best, and expected returns for each subclass
        Worst = capital * (class2Pct / 100) * (SubPcts(i) / 100) * SubMinRates(i)
        Best = capital * (class2Pct / 100) * (SubPcts(i) / 100) * SubMaxRates(i)
        Expected = (Worst + Best) / 2

        ' Add Sub values to Class 2 totals
        Class2TotalExpected = Class2TotalExpected + Expected
        Class2TotalBest = Class2TotalBest + Best
        Class2TotalWorst = Class2TotalWorst + Worst

        ' Construct the result message for each Sub
        resultMessage = resultMessage & "  " & SubNames(i) & ":" & vbCrLf & _
                        "    Expected: " & IIf(Expected < 0, "(" & FormatNumber(Abs(Expected), 3) & ")",
FormatNumber(Expected, 3)) & " billion." & vbCrLf & _
                        "    Best: " & IIf(Best < 0, "(" & FormatNumber(Abs(Best), 3) & ")",
FormatNumber(Best, 3)) & " billion." & vbCrLf & _
                        "    Worst: " & IIf(Worst < 0, "(" & FormatNumber(Abs(Worst), 3) & ")",
FormatNumber(Worst, 3)) & " billion." & vbCrLf

        ' Record the expected value in cells E, I, M (adjusted for the starting column)
        ws.Cells(startRow, 5 + (i - 1) * 4).Value = Expected
        ws.Cells(startRow, 6 + (i - 1) * 4).Value = Best
        ws.Cells(startRow, 7 + (i - 1) * 4).Value = Worst
    Next i

    ' Construct the result message for Class 2
    resultMessage = "RETURN CASE FOR THE CURRENT YEAR->" & vbCrLf & _
                    "  Expected: " & FormatNumber(Class2TotalExpected, 3) & " billion." & vbCrLf & _
                    "  Best: " & FormatNumber(Class2TotalBest, 3) & " billion." & vbCrLf & _
                    "  Worst: " & FormatNumber(Class2TotalWorst, 3) & " billion." & vbCrLf & _
                    " Sub Class->" & vbCrLf & resultMessage

    ' Update overall totals for all classes
    TotalExpectedAllClasses = TotalExpectedAllClasses + Class2TotalExpected
    TotalBestAllClasses = TotalBestAllClasses + Class2TotalBest
    TotalWorstAllClasses = TotalWorstAllClasses + Class2TotalWorst

    ' Display the result in one message box
    MsgBox resultMessage, vbInformation
End Sub


Sub Class3Return(class3Pct As Double)
    ' Declare variables
    Dim capital As Double
    Dim year As Double
    Dim SubPcts(1 To 3) As Double
    Dim SubMinRates(1 To 3) As Double
    Dim SubMaxRates(1 To 3) As Double
    Dim SubNames(1 To 3) As String
    Dim resultMessage As String
    Dim Class3TotalExpected As Double
    Dim Class3TotalBest As Double
    Dim Class3TotalWorst As Double

    ' Input total capital and year
    capital = currentCapital
    year = numYears

    ' Set reference to the "Class 3 Details" worksheet
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Class 3 Details")

    ' Find the last used row in column A of "Class 3 Details"
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    ' Increment lastRow by 1 to get the startRow
    Dim startRow As Long
    startRow = lastRow + 1

    ' Store the year, capital, and class3Pct in the "Class 3 Details" worksheet
    ws.Cells(startRow, 1).Value = year
    ws.Cells(startRow, 2).Value = capital
    ws.Cells(startRow, 3).Value = class3Pct

    ' Set Sub allocations and rates for Class 3
    SubNames(1) = "GSC"
    SubMinRates(1) =-0.05
    SubMaxRates(1) = 0.05
```

```
   SubNames(2) = "ENERGY"
   SubMinRates(2) = 0
   SubMaxRates(2) = 0.08

   SubNames(3) = "AUTOMATIVE"
   SubMinRates(3) =-0.1
   SubMaxRates(3) = 0.05

   ' Prompt the user for SubPcts for Class 3
   Dim i As Integer
   For i = 1 To 3
      Do
         SubPcts(i) = InputBox("Enter the % for " & SubNames(i) & vbCrLf & "Remaining: " &
FormatNumber(100- SubPcts(1)- SubPcts(2)- SubPcts(3), 2) & "%", "Class 3")
         If SubPcts(1) + SubPcts(2) + SubPcts(3) > 100 Then
            MsgBox "Total percentage exceeds 100%. Please re-enter.", vbExclamation
         End If
      Loop While SubPcts(1) + SubPcts(2) + SubPcts(3) > 100

      ' Record SubPcts in cells D, H, L (adjusted for the starting column)
      ws.Cells(startRow, 4 + (i - 1) * 4).Value = SubPcts(i)
   Next i

   ' Loop through Subs to calculate returns
   For i = 1 To 3
      Dim Worst As Double
      Dim Best As Double
      Dim Expected As Double

      ' Calculate worst, best, and expected returns for each subclass
      Worst = capital * (class3Pct / 100) * (SubPcts(i) / 100) * SubMinRates(i)
      Best = capital * (class3Pct / 100) * (SubPcts(i) / 100) * SubMaxRates(i)
      Expected = (Worst + Best) / 2

      ' Add Sub values to Class 3 totals
      Class3TotalExpected = Class3TotalExpected + Expected
      Class3TotalBest = Class3TotalBest + Best
      Class3TotalWorst = Class3TotalWorst + Worst

      ' Construct the result message for each Sub
      resultMessage = resultMessage & "  " & SubNames(i) & ":" & vbCrLf & _
                  "    Expected: " & IIf(Expected < 0, "(" & FormatNumber(Abs(Expected), 3) & ")",
FormatNumber(Expected, 3)) & " billion." & vbCrLf & _
                  "    Best: " & IIf(Best < 0, "(" & FormatNumber(Abs(Best), 3) & ")",
FormatNumber(Best, 3)) & " billion." & vbCrLf & _
                  "    Worst: " & IIf(Worst < 0, "(" & FormatNumber(Abs(Worst), 3) & ")",
FormatNumber(Worst, 3)) & " billion." & vbCrLf
```

```
        ' Record the expected value in cells E, I, M (adjusted for the starting column)
        ws.Cells(startRow, 5 + (i - 1) * 4).Value = Expected
        ws.Cells(startRow, 6 + (i - 1) * 4).Value = Best
        ws.Cells(startRow, 7 + (i - 1) * 4).Value = Worst
    Next i

    ' Construct the result message for Class 3
    resultMessage = "CLASS 3 RETURN CASE FOR THE CURRENT YEAR->" & vbCrLf & _
                  "  Expected: " & IIf(Class3TotalExpected < 0, "(" &
FormatNumber(Abs(Class3TotalExpected), 3) & ")", FormatNumber(Class3TotalExpected, 3))
& " billion." & vbCrLf & _
                  "  Best: " & FormatNumber(Class3TotalBest, 3) & " billion." & vbCrLf & _
                  "  Worst: " & FormatNumber(Class3TotalWorst, 3) & " billion." & vbCrLf & _
                  " Sub Class->" & vbCrLf & resultMessage

    ' Update overall totals for all classes
    TotalExpectedAllClasses = TotalExpectedAllClasses + Class3TotalExpected
    TotalBestAllClasses = TotalBestAllClasses + Class3TotalBest
    TotalWorstAllClasses = TotalWorstAllClasses + Class3TotalWorst

    ' Display the result in one message box
    MsgBox resultMessage, vbInformation
End Sub


Sub Class4Return(class4Pct As Double)
    ' Declare variables
    Dim capital As Double
    Dim year As Double
    Dim resultMessage As String

    ' Input total capital and year
    capital = currentCapital
    year = numYears

    ' Set reference to the "Class 4 Details" worksheet
    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("Class 4 Details")

    ' Find the last used row in column A of "Class 4 Details"
    Dim lastRow As Long
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    ' Increment lastRow by 1 to get the startRow
    Dim startRow As Long
    startRow = lastRow + 1
    ' Store the year, capital, and class4Pct in the "Class 4 Details" worksheet
    ws.Cells(startRow, 1).Value = year
    ws.Cells(startRow, 2).Value = capital
    ws.Cells(startRow, 3).Value = class4Pct

    ' Prompt the user for the interest rates for the current year
    Dim minRate As Double
    Dim maxRate As Double
    minRate = InputBox("Enter the MIN interest rate for the current year:", "FD Banks")
    maxRate = InputBox("Enter the MAX interest rate for the current year:", "FD Banks")
    ws.Cells(startRow, 4).Value = minRate
    ws.Cells(startRow, 5).Value = maxRate

    ' Calculate returns for the current year
    Dim YearExpected As Double
    Dim YearBest As Double
    Dim YearWorst As Double

    YearExpected = capital * (class4Pct / 100) * (((minRate + maxRate) / 100) / 2)
    YearBest = capital * (class4Pct / 100) * (maxRate / 100)
    YearWorst = capital * (class4Pct / 100) * (minRate / 100)

    ' Construct the result message for the current year
    resultMessage = "FIXED DEPOSITS RETURN CASE FOR THE CURRENT YEAR->" & vbCrLf & _
                "  Expected: " & IIf(YearExpected < 0, "(" & FormatNumber(Abs(YearExpected), 3)
& ")", FormatNumber(YearExpected, 3)) & " billion." & vbCrLf & _
                "  Best: " & IIf(YearBest < 0, "(" & FormatNumber(Abs(YearBest), 3) & ")",
FormatNumber(YearBest, 3)) & " billion." & vbCrLf & _
                "  Worst: " & IIf(YearWorst < 0, "(" & FormatNumber(Abs(YearWorst), 3) & ")",
FormatNumber(YearWorst, 3)) & " billion." & vbCrLf

    ' Record the calculated values in cells F, G, H
    ws.Cells(startRow, 6).Value = YearExpected
    ws.Cells(startRow, 7).Value = YearBest
    ws.Cells(startRow, 8).Value = YearWorst

    ' Add Class 4 values to the global totals for all classes
    TotalExpectedAllClasses = TotalExpectedAllClasses + YearExpected
    TotalBestAllClasses = TotalBestAllClasses + YearBest
    TotalWorstAllClasses = TotalWorstAllClasses + YearWorst

    ' Display the result for the current year in one message box
    MsgBox resultMessage, vbInformation
End Sub
```

```
' Clears the specified range in the "Annual Investment Record" worksheet
Sub ClearTable()
   Dim ws As Worksheet
   Set ws = ThisWorkbook.Sheets("Annual Investment Record")

   ' Clear the specified range in the worksheet
   ws.Range("A6:E100").ClearContents
End Sub

' Clears the specified range in the "Class 1 Details" worksheet
Sub ClearTable1()
   Dim ws As Worksheet
   Set ws = ThisWorkbook.Sheets("Class 1 Details")

   ' Clear the specified range in the worksheet
   ws.Range("A8:O100").ClearContents
End Sub

' Clears the specified range in the "Class 2 Details" worksheet
Sub ClearTable2()
   Dim ws As Worksheet
   Set ws = ThisWorkbook.Sheets("Class 2 Details")

   ' Clear the specified range in the worksheet
   ws.Range("A8:O100").ClearContents
End Sub

' Clears the specified range in the "Class 3 Details" worksheet
Sub ClearTable3()
   Dim ws As Worksheet
   Set ws = ThisWorkbook.Sheets("Class 3 Details")

   ' Clear the specified range in the worksheet
   ws.Range("A8:O100").ClearContents
End Sub

' Clears the specified range in the "Class 4 Details" worksheet
Sub ClearTable4()
   Dim ws As Worksheet
   Set ws = ThisWorkbook.Sheets("Class 4 Details")

   ' Clear the specified range in the worksheet
   ws.Range("A8:H100").ClearContents
End Sub
```