Documentation of VBExercises project

Kaustuv Prajapati, MBA IT 6th batch

Project Github link: <https://github.com/kaustuv-praz/VBExercises>

Table of Contents

[Documentation of VBExercises project 3](#_Toc71628558)

[Disclaimer 3](#_Toc71628559)

[General Application 4](#_Toc71628560)

[Exercise 1 5](#_Toc71628561)

[Core Code 5](#_Toc71628562)

[Click Handler Code 6](#_Toc71628563)

[Exercise 2 7](#_Toc71628564)

[Core Code 7](#_Toc71628565)

[Button Click Handler 8](#_Toc71628566)

[Exercise 3 9](#_Toc71628567)

[Core Code 10](#_Toc71628568)

[Button Click Handler 10](#_Toc71628569)

[Exercise 4 11](#_Toc71628570)

[Core Code 12](#_Toc71628571)

[Button Handler code 12](#_Toc71628572)

[Exercise 5 13](#_Toc71628573)

[Core code 14](#_Toc71628574)

[Button click Handler Code 15](#_Toc71628575)

[Exercise 6 16](#_Toc71628576)

[Core Code 17](#_Toc71628577)

[Exercise 7 18](#_Toc71628578)

[Core code 18](#_Toc71628579)

[Button Handler Code 19](#_Toc71628580)

[Exercise 8 20](#_Toc71628581)

[Core code 21](#_Toc71628582)

[TextChanged Handler 22](#_Toc71628583)

[Exercise 9 23](#_Toc71628584)

[Main Code 24](#_Toc71628585)

[Exercise 10 25](#_Toc71628586)

[Main code 26](#_Toc71628587)

[Exercise 11 27](#_Toc71628588)

[Core code 28](#_Toc71628589)

[Exercise 12 29](#_Toc71628590)

[Core Code 30](#_Toc71628591)

# Documentation of VBExercises project

## Disclaimer

This doc explains how the code structure is implemented in the Application. Explains about the components used.

Also contains the note for various techniques and tools, example, syntax, good practices, shortcuts, custom methods and code flow.

# General Application

Figure 1 App UI

General MenuStrip is added for general functions.

TabControl is added for multi tabl selection

Exercises tab:

* ListView for exercise selection
* Panel for exercise Form display.

# Exercise 1

Write a program to calculate discount on the basis of following assumption: a) If purchased amount is greater than or equal to 1000, discount is 5%

## Core Code

Figure 2 Exercise 1 in action. Calculating discount

Below is the core CODE component

Private Sub Update\_Discount\_Settings(newPurchaseAmt)

If newPurchaseAmt >= 1000 Then

discountable = True

DisStatLvlVal.Text = "YES"

DisStatLvlVal.ForeColor = Color.Green

DisPerLblVal.Text = "5%"

Else

discountable = False

DisStatLvlVal.Text = "NO"

DisStatLvlVal.ForeColor = Color.Red

DisPerLblVal.Text = "Na"

End If

End Sub

## Click Handler Code

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CalculateBtn.Click

' CALCULATING 5% DISCOUNT ON Purchase Amt

Dim discountAmt, disPercent As Double

If discountable Then

disPercent = 0.05 ' in percentage

discountAmt = Math.Round(newPurchaseAmt \* disPercent, 2)

newPurchaseAmt = Math.Round(newPurchaseAmt - discountAmt, 2)

Else

discountAmt = 0

End If

DisAmtValLbl.Text = discountAmt & " Rs /-"

FinalAmtLblVal.Text = newPurchaseAmt & " Rs /-"

End Sub

# Exercise 2

Write a program to calculate discount on the basis of following assumption: a) If purchased amount is greater than or equal to 1000, discount is 5% b) If purchased amount is less than 1000, discount is 3%

## Core Code

Private Sub Update\_Discount\_Settings(newPurchaseAmt)

If newPurchaseAmt >= 1000 Then

discountable = True

DisStatLvlVal.Text = "YES"

DisStatLvlVal.ForeColor = Color.Green

DisPerLblVal.Text = "5%"

disPercent = 0.05 ' in percentage

ElseIf newPurchaseAmt < 1000 Then

discountable = True

DisStatLvlVal.Text = "YES"

DisStatLvlVal.ForeColor = Color.Green

DisPerLblVal.Text = "3%"

disPercent = 0.03 ' in percentage

End If

End Sub

Figure 3 Exercise 2 in action

## Button Click Handler

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CalculateBtn.Click

Dim discountAmt As Double

If discountable Then

discountAmt = Math.Round(newPurchaseAmt \* disPercent, 2)

newPurchaseAmt = Math.Round(newPurchaseAmt - discountAmt, 2)

Else

discountAmt = 0

End If

DisAmtValLbl.Text = discountAmt & " Rs /-"

FinalAmtLblVal.Text = newPurchaseAmt & " Rs /-"

End Sub

# Exercise 3

Write a program to calculate discount on the basis of following assumption: a)

If purchased amount is greater than or equal to 5000, discount is 10% b)

If purchased amount is greater than or equal to 4000 and less than 5000, discount is 7% c)

If purchased amount is greater than or equal to 3000 and less than 4000, discount is 5% d)

If purchased amount is greater than or equal to 2000 and less than 3000, discount is 3% e)

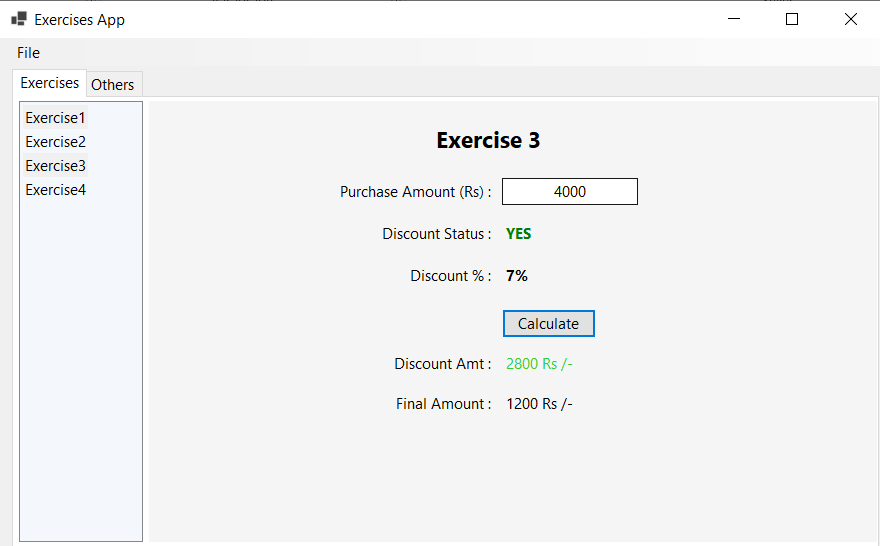
 If purchased amount is less than 2000, discount is 2%

Figure 4 Exercise 3 in action

## Core Code

Private Sub Update\_Discount\_Settings(newPurchaseAmt)

If newPurchaseAmt >= 5000 Then

Calculate\_Discount(True, "YES", Color.Green, "10%", 0.1)

ElseIf newPurchaseAmt >= 4000 And newPurchaseAmt < 5000 Then

Calculate\_Discount(True, "YES", Color.Green, "7%", 0.07)

disPercent = 0.7 ' in percentage

ElseIf newPurchaseAmt >= 3000 And newPurchaseAmt < 4000 Then

Calculate\_Discount(True, "YES", Color.Green, "5%", 0.05)

ElseIf newPurchaseAmt >= 2000 And newPurchaseAmt < 3000 Then

Calculate\_Discount(True, "YES", Color.Green, "3%", 0.03)

ElseIf newPurchaseAmt < 2000 Then

Calculate\_Discount(True, "YES", Color.Green, "2%", 0.02)

End If

End Sub

Private Sub Calculate\_Discount(discountableBool, disStatLblVal, disStatLblColor, disPerLblValNew, disPercentVal)

discountable = discountableBool

Me.DisStatLblVal.Text = disStatLblVal

Me.DisStatLblVal.ForeColor = disStatLblColor

DisPerLblVal.Text = disPerLblValNew

disPercent = disPercentVal ' in percentage

End Sub

## Button Click Handler

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CalculateBtn.Click

Dim discountAmt As Double

If discountable Then

discountAmt = Math.Round(newPurchaseAmt \* disPercent, 2)

newPurchaseAmt = Math.Round(newPurchaseAmt - discountAmt, 2)

Else

discountAmt = 0

End If

DisAmtValLbl.Text = discountAmt & " Rs /-"

FinalAmtLblVal.Text = newPurchaseAmt & " Rs /-"

End Sub

# Exercise 4

Write a program to calculate the simple interest on the basis of following assumption:

a) If balance is greater than 99999, interest is 7 %

b) If balance is greater than or equal to 50000 and less than 100000 interest is 5 %

c) If balance is less than 50000, interest is 3%

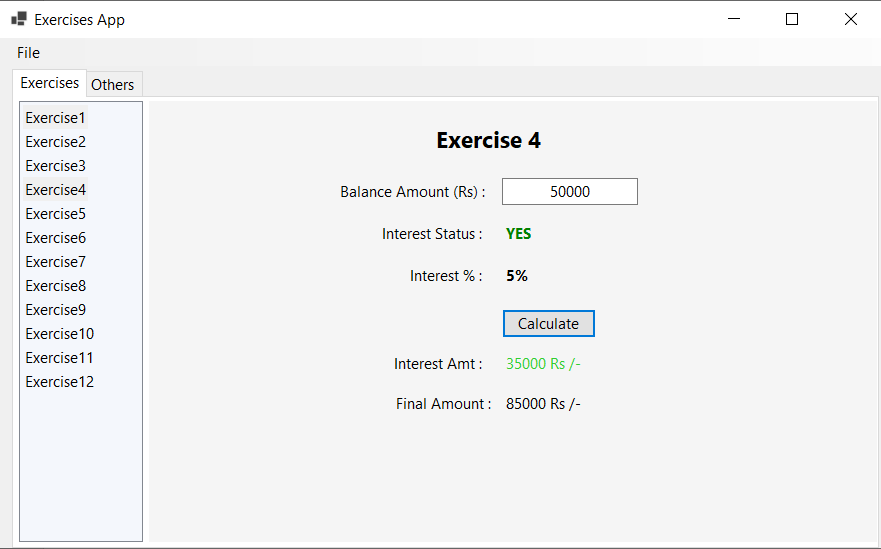


Figure 5 Exercise 4 code in action

## Core Code

Private Sub Update\_Discount\_Settings(newBalanceAmt)

If newBalanceAmt > 99999 Then

Calculate\_Discount(True, "YES", Color.Green, "7%", 0.07)

ElseIf newBalanceAmt >= 50000 And newBalanceAmt < 100000 Then

Calculate\_Discount(True, "YES", Color.Green, "5%", 0.05)

interestPercent = 0.7 ' in percentage

ElseIf newBalanceAmt < 50000 Then

Calculate\_Discount(True, "YES", Color.Green, "3%", 0.03)

End If

End Sub

Private Sub Calculate\_Discount(discountableBool, disStatLblVal, disStatLblColor, disPerLblValNew, disPercentVal)

interestable = discountableBool

Me.InterestStatLblVal.Text = disStatLblVal

Me.InterestStatLblVal.ForeColor = disStatLblColor

InterestPerLblVal.Text = disPerLblValNew

interestPercent = disPercentVal ' in percentage

End Sub

## Button Handler code

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CalculateBtn.Click

' CALCULATING 5% DISCOUNT ON Purchase Amt

Dim interestAmt As Double

If interestable Then

interestAmt = Math.Round(newBalanceAmt \* interestPercent, 2)

newBalanceAmt = Math.Round(newBalanceAmt + interestAmt, 2)

Else

interestAmt = 0

End If

InterestAmtValLbl.Text = interestAmt & " Rs /-"

FinalAmtLblVal.Text = newBalanceAmt & " Rs /-"

End Sub

# Exercise 5

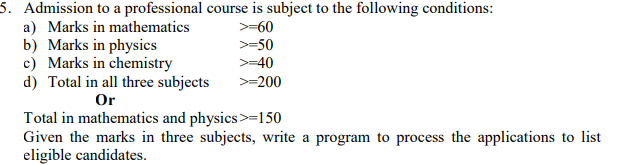
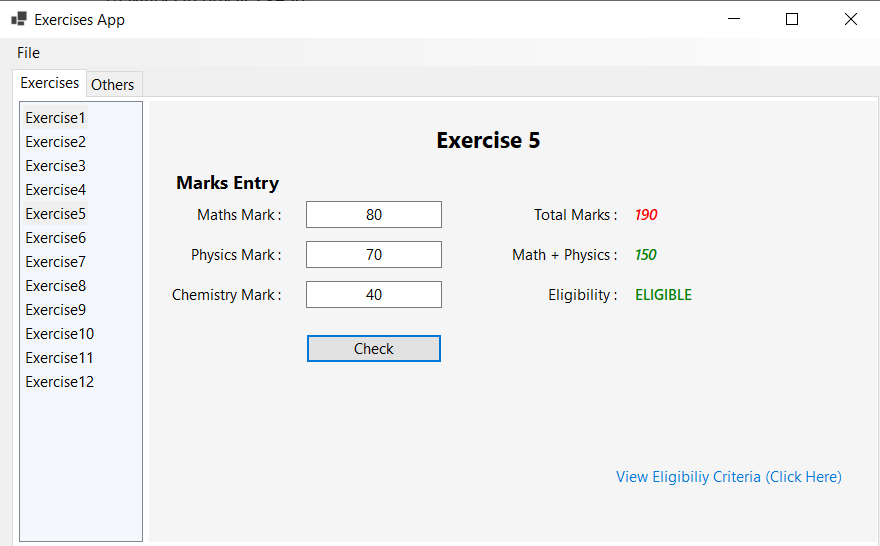


Figure 6 Exercise 5 in acction

## Core code

Private Sub Update\_Discount\_Settings(newBalanceAmt)

If newBalanceAmt > 99999 Then

Calculate\_Discount(True, "YES", Color.Green, "7%", 0.07)

ElseIf newBalanceAmt >= 50000 And newBalanceAmt < 100000 Then

Calculate\_Discount(True, "YES", Color.Green, "5%", 0.05)

interestPercent = 0.7 ' in percentage

ElseIf newBalanceAmt < 50000 Then

Calculate\_Discount(True, "YES", Color.Green, "3%", 0.03)

End If

End Sub

Private Sub Calculate\_Discount(discountableBool, disStatLblVal, disStatLblColor, disPerLblValNew, disPercentVal)

interestable = discountableBool

Me.TotalMarkLblVal.Text = disStatLblVal

Me.TotalMarkLblVal.ForeColor = disStatLblColor

TotalMPMarkLblVal.Text = disPerLblValNew

interestPercent = disPercentVal ' in percentage

End Sub

## Button click Handler Code

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CheckBtn.Click

' EVALUATE ELIGIBILITY

Dim mMark, pMark, cMark, totalMark, totalMPMark As Double

mMark = MathsMarkTB.Text

pMark = PhysicsMarkTB.Text

cMark = ChemistryMarkTB.Text

totalMark = mMark + pMark + cMark

totalMPMark = mMark + pMark

TotalMarkLblVal.Text = totalMark

TotalMPMarkLblVal.Text = totalMPMark

If (mMark >= 60 And pMark >= 50 And cMark >= 40 And totalMark >= 200) Or (totalMPMark >= 150) Then

If mMark >= 60 And pMark >= 50 And cMark >= 40 And totalMark >= 200 Then

TotalMarkLblVal.ForeColor = Color.Green

Else

TotalMarkLblVal.ForeColor = Color.Red

End If

If totalMPMark >= 150 Then

TotalMPMarkLblVal.ForeColor = Color.Green

Else

TotalMPMarkLblVal.ForeColor = Color.Red

End If

EligibiltyLblVal.Text = "ELIGIBLE"

EligibiltyLblVal.ForeColor = Color.Green

Else

TotalMarkLblVal.ForeColor = Color.Red

TotalMPMarkLblVal.ForeColor = Color.Red

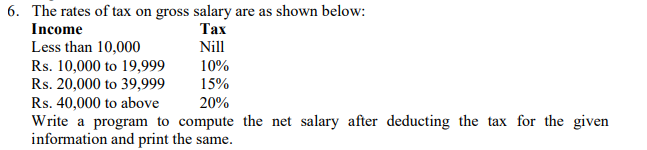
EligibiltyLblVal.Text = "NOT ELIGIBLE"

EligibiltyLblVal.ForeColor = Color.Red

End If

End Sub

# Exercise 6



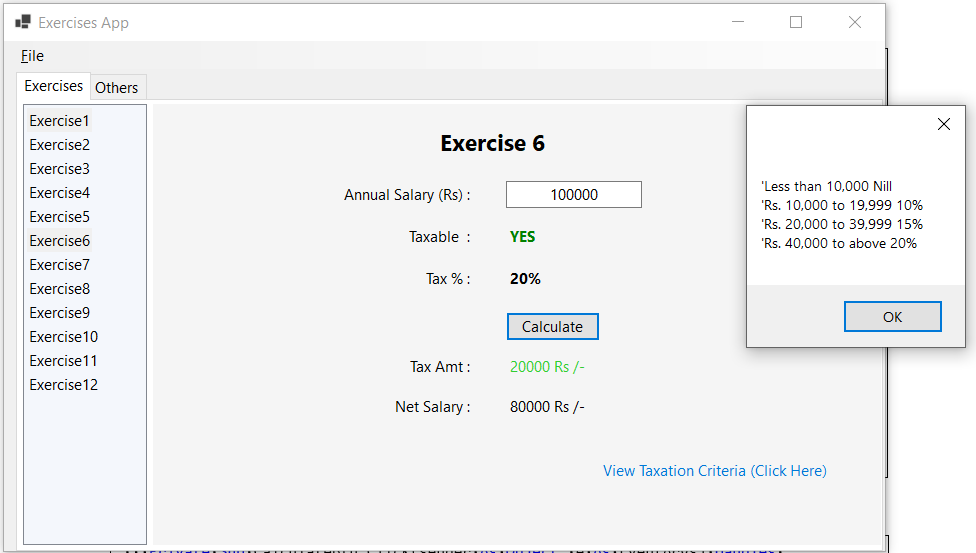


Figure 7 Exercise 6 in action

## Core Code

Private Sub Update\_Tax\_Settings(newAnnSalaryAmt)

If newAnnSalaryAmt >= 40000 Then

Calculate\_Tax(True, "YES", Color.Green, "20%", 0.2)

ElseIf newAnnSalaryAmt >= 20000 And newAnnSalaryAmt < 40000 Then

Calculate\_Tax(True, "YES", Color.Green, "15%", 0.15)

ElseIf newAnnSalaryAmt >= 10000 And newAnnSalaryAmt < 20000 Then

Calculate\_Tax(True, "YES", Color.Green, "10%", 0.1)

ElseIf newAnnSalaryAmt < 10000 Then

Calculate\_Tax(False, "NO", Color.Red, "Na", 0)

End If

End Sub

Private Sub Calculate\_Tax(taxableBool, taxStatLblVal, taxStatLblColor, taxPerLblValNew, taxPercentVal)

taxable = taxableBool

Me.TaxableLblVal.Text = taxStatLblVal

Me.TaxableLblVal.ForeColor = taxStatLblColor

TaxPerLblVal.Text = taxPerLblValNew

taxPercent = taxPercentVal ' in percentage

End Sub

Private Sub EligibilityCriteriaLbl\_Click(sender As Object, e As EventArgs) Handles EligibilityCriteriaLbl.Click

MessageBox.Show("'Less than 10,000 Nill

'Rs. 10,000 to 19,999 10%

'Rs. 20,000 to 39,999 15%

'Rs. 40,000 to above 20%

")

End Sub

Button Handler code

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CalculateBtn.Click

Dim taxAmt As Double

If taxable Then

taxAmt = Math.Round(newAnnualSalaryAmt \* taxPercent, 2)

newAnnualSalaryAmt = Math.Round(newAnnualSalaryAmt - taxAmt, 2)

Else

taxAmt = 0

TaxPerLblVal.ForeColor = Color.Red

End If

TaxAmtValLbl.Text = taxAmt & " Rs /-"

FinalAmtLblVal.Text = newAnnualSalaryAmt & " Rs /-"

End Sub

# Exercise 7



## Core code

Figure 8 Exercise 7 in action

Private Sub Update\_Commission\_Settings(newMSalesAmt)

If newMSalesAmt < 10000 Then

Calculate\_Commission("5%", 0.05)

Else

Calculate\_Commission("10%", 0.1)

End If

End Sub

Private Sub Calculate\_Commission(commissionPerLblValNew, commissionPercentVal)

CommissionPerLblVal.Text = commissionPerLblValNew

commissionPercent = commissionPercentVal ' in percentage

End Sub

Private Sub EligibilityCriteriaLbl\_Click(sender As Object, e As EventArgs) Handles EligibilityCriteriaLbl.Click

MessageBox.Show("'Jet Company gives 5% commission To its salesman If their monthly sales are less than Rs.

'10,000 And a 10% commission if it Is equal to Or greater than Rs. 10,000. Write a program

'to calculate commission at the end of the month.")

End Sub

## Button Handler Code

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CalculateBtn.Click

Dim comissionAmt, finalPaycheckAmt As Double

comissionAmt = Math.Round(newMonthlySalesAmt \* commissionPercent, 2)

finalPaycheckAmt = Math.Round(newMonthlySalesAmt + comissionAmt, 2)

DisAmtValLbl.Text = comissionAmt & " Rs /-"

FinalAmtLblVal.Text = finalPaycheckAmt & " Rs /-"

End Sub

# Exercise 8

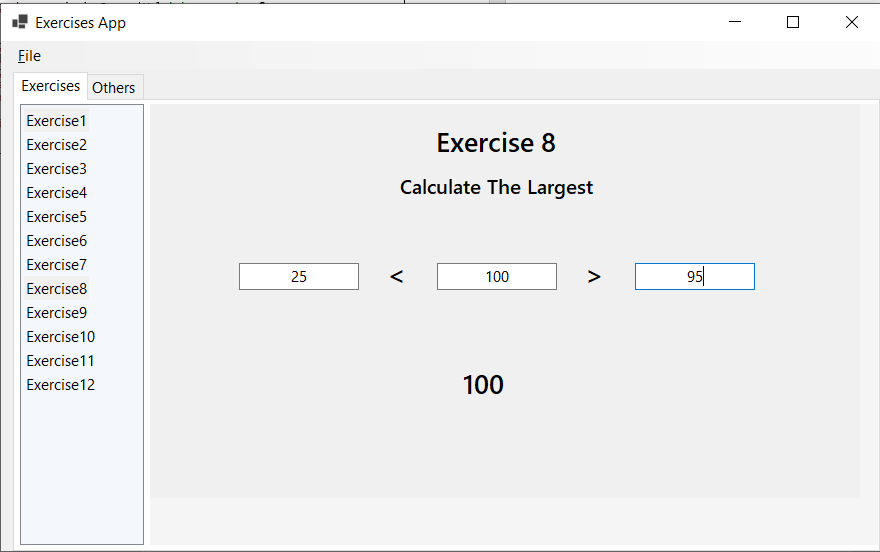
8. Write a program to print the largest number among three numbers input by the user.

Figure 9 Exercise 8 in action

## Core code

Private Sub Calculate\_Largest()

Dim firstNum, secondNum, thirdNum As Double

Double.TryParse(FirstNumberTB.Text, firstNum)

Double.TryParse(SecondNumberTB.Text, secondNum)

Double.TryParse(ThirdNumberTB.Text, thirdNum)

If firstNum > secondNum And firstNum > thirdNum Then

' First number is grater

FirstIndicatorLbl.Text = ">"

If secondNum > thirdNum Then

SecondIndicatorLbl.Text = ">"

Else

SecondIndicatorLbl.Text = "<"

End If

LargetNumberLblVal.Text = firstNum

ElseIf secondNum > firstNum And secondNum > thirdNum Then

' Second number is greater

FirstIndicatorLbl.Text = "<"

SecondIndicatorLbl.Text = ">"

LargetNumberLblVal.Text = secondNum

Else

' Third Number is greater

SecondIndicatorLbl.Text = "<"

If firstNum > secondNum Then

FirstIndicatorLbl.Text = ">"

Else

FirstIndicatorLbl.Text = "<"

End If

LargetNumberLblVal.Text = " " & thirdNum

End If

End Sub

## TextChanged Handler

Private Sub FirstNumberTB\_TextChanged(sender As Object, e As EventArgs) Handles FirstNumberTB.TextChanged

Calculate\_Largest()

End Sub

Private Sub SecondNumberTB\_TextChanged(sender As Object, e As EventArgs) Handles SecondNumberTB.TextChanged

Calculate\_Largest()

End Sub

Private Sub ThirdNumberTB\_TextChanged(sender As Object, e As EventArgs) Handles ThirdNumberTB.TextChanged

Calculate\_Largest()

End Sub

# Exercise 9

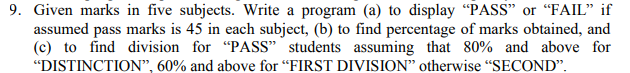
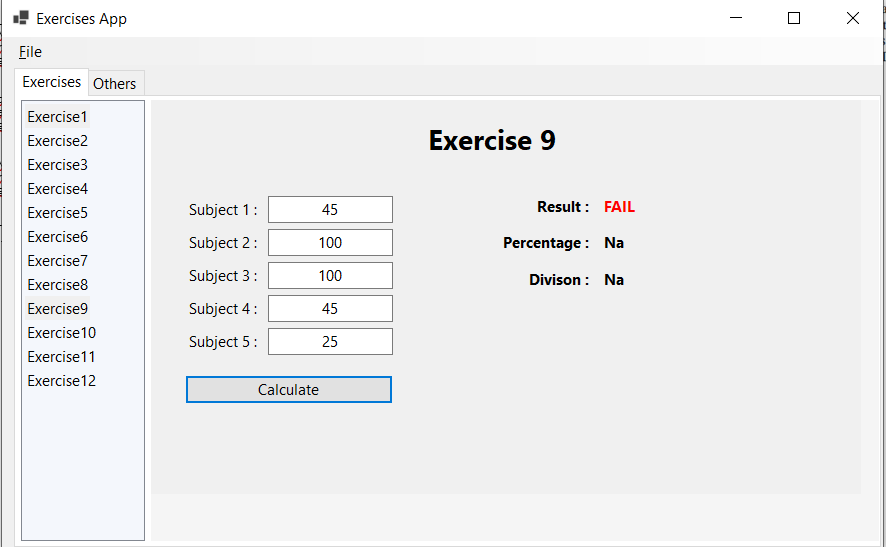


Figure 10 Exercise 9 in action

## Main Code

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CalculateBtn.Click

Dim sub1, sub2, sub3, sub4, sub5 As Double

Dim resultStatus As Boolean

Double.TryParse(Subject1Tb.Text, sub1)

Double.TryParse(Subject2Tb.Text, sub2)

Double.TryParse(Subject3Tb.Text, sub3)

Double.TryParse(Subject4Tb.Text, sub4)

Double.TryParse(Subject5Tb.Text, sub5)

If sub1 < 45 Or sub2 < 45 Or sub3 < 45 Or sub4 < 45 Or sub5 < 45 Then

ResultStatusLbl.Text = "FAIL"

ResultStatusLbl.ForeColor = Color.Red

resultStatus = False

Else

ResultStatusLbl.Text = "PASS"

ResultStatusLbl.ForeColor = Color.Green

resultStatus = True

End If

If resultStatus Then

Dim totalMarks, percentage As Double

Dim division As String

totalMarks = sub1 + sub2 + sub3 + sub4 + sub5

percentage = totalMarks / 5

PercentageLblVal.Text = percentage

If percentage > 80 Then

division = "DISTINCTION"

ElseIf percentage > 60 Then

division = "FIRST DIVISION"

Else

division = "SECOND DIVISION"

End If

DivisionLblVal.Text = division

Else

PercentageLblVal.Text = "Na"

DivisionLblVal.Text = "Na"

End If

End Sub

# Exercise 10

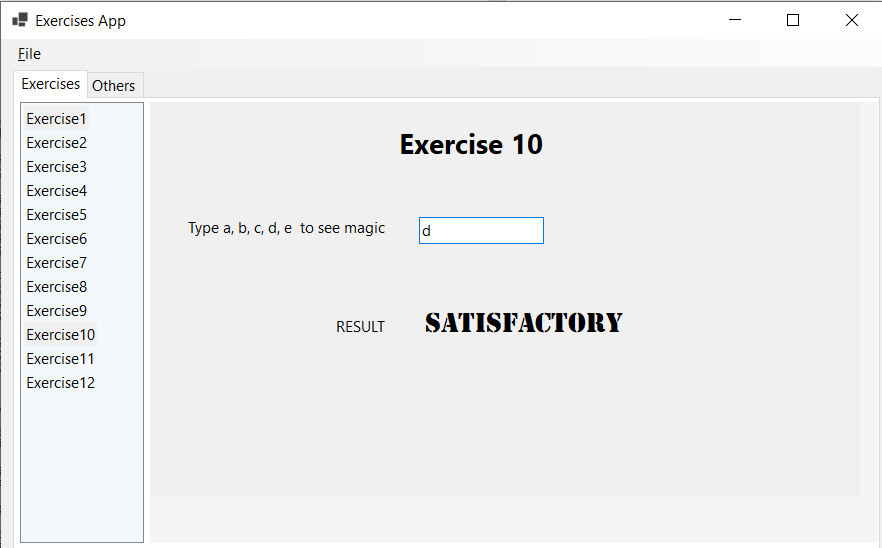
10. Write a program to display EXCELLENT, VERY GOOD, GOOD, SATISFACTORY, or FAIL if the user enters A, B, C, D, or E respectively

Figure 11 Exercise 10 in action

## Main code

Private Sub UserInputTb\_TextChanged(sender As Object, e As EventArgs) Handles UserInputTb.TextChanged

Dim newChar As Char

Dim userInput As String

userInput = UserInputTb.Text

If userInput.Length > 0 Then

'MessageBox.Show(newChar)

newChar = userInput(userInput.Length - 1)

Dim newCharString As String

newCharString = newChar.ToString

UserInputTb.Text = newCharString

UserInputTb.SelectionStart = UserInputTb.Text.Length

UserInputTb.SelectionLength = 0

If newCharString = "a" Or newCharString = "A" Then

ResultLblVal.Text = "EXCELLENT"

ElseIf newCharString = "b" Or newCharString = "B" Then

ResultLblVal.Text = "VERY GOOD"

ElseIf newCharString = "c" Or newCharString = "C" Then

ResultLblVal.Text = "GOOD"

ElseIf newCharString = "d" Or newCharString = "D" Then

ResultLblVal.Text = "SATISFACTORY"

ElseIf newCharString = "e" Or newCharString = "E" Then

ResultLblVal.Text = "FAIL"

Else

ResultLblVal.Text = "Na"

End If

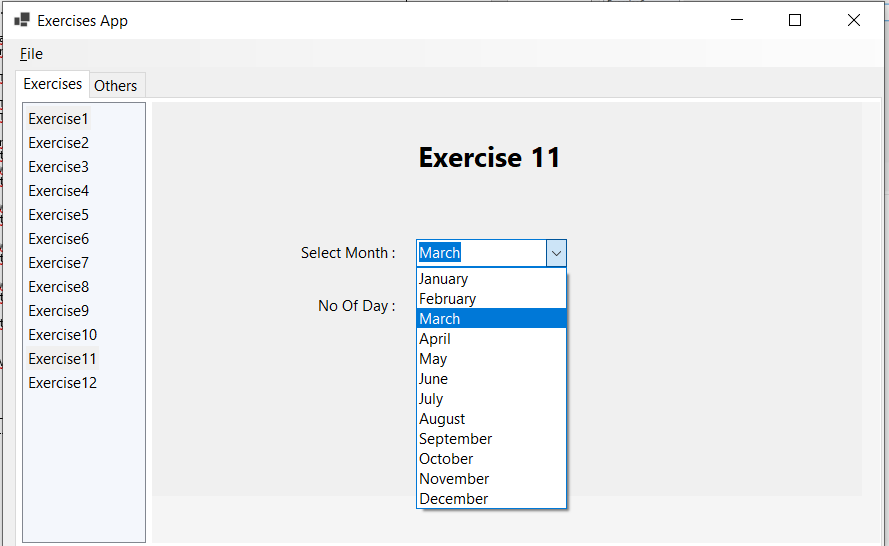
Else

ResultLblVal.Text = "Na"

End If

End Sub

# Exercise 11

11. Write a program to display number of days in a month.

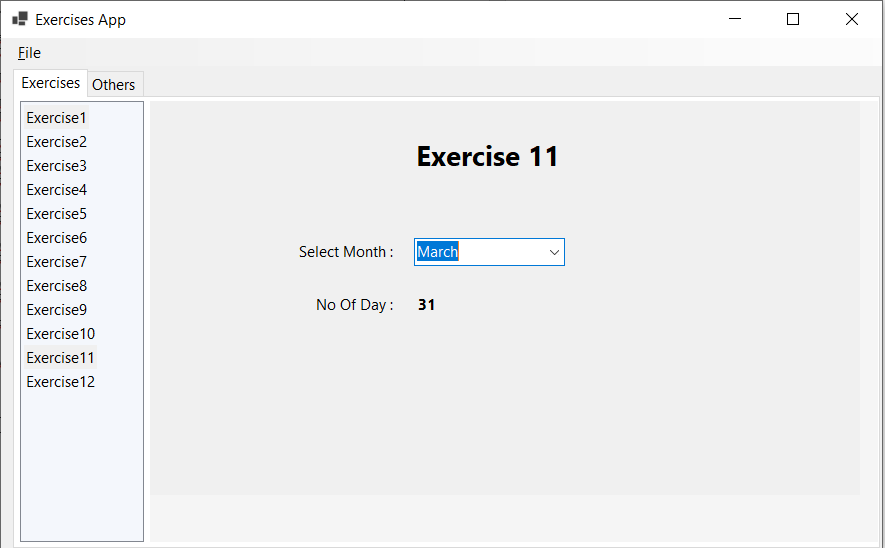


Figure 12 Exercise 11 in action

## Core code

Dim monthArray() As String = {

"January",

"February",

"March",

"April",

"May ",

"June ",

"July ",

"August ",

"September",

"October",

"November",

"December"

}

Dim noOfDaysArray() As Int64 = {

31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31

}

Private Sub Exercise11\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

MonthSelectorCb.DisplayMember = "Text"

MonthSelectorCb.ValueMember = "Value"

Dim tableData As New DataTable

tableData.Columns.Add("Text", GetType(String))

tableData.Columns.Add("Value", GetType(Int64))

For i = 0 To 11

tableData.Rows.Add(monthArray(i), noOfDaysArray(i))

Next

MonthSelectorCb.DataSource = tableData

End Sub

Private Sub MonthSelectorCb\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles MonthSelectorCb.SelectedIndexChanged

NoOfDaysLblVal.Text = MonthSelectorCb.SelectedValue

End Sub

# Exercise 12

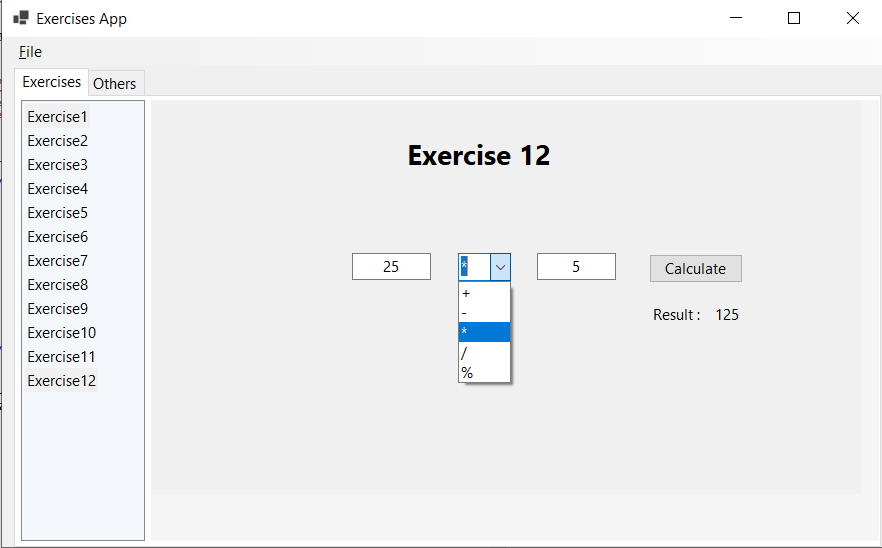
12. Write a program using select case statement to develop a simple calculator for +, -, \*, /, and % operators.

Figure 13 Exercise 12 in action

## Core Code

Public Class Exercise12

Dim operatorSign As String

Private Sub CalculateBtn\_Click(sender As Object, e As EventArgs) Handles CalculateBtn.Click

Dim firstNum, secondNum As Double

Dim resultData As Double

Double.TryParse(FirstNumberTb.Text, firstNum)

Double.TryParse(secondNumberTb.Text, secondNum)

Select Case operatorSign

Case "+"

resultData = firstNum + secondNum

Case "-"

resultData = firstNum - secondNum

Case "\*"

resultData = firstNum \* secondNum

Case "/"

resultData = firstNum / secondNum

Case "%"

resultData = firstNum Mod secondNum

End Select

ResultLblVal.Text = resultData

End Sub

Private Sub OperatorCb\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles OperatorCb.SelectedIndexChanged

operatorSign = OperatorCb.SelectedItem.ToString

End Sub

Private Sub Exercise12\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

OperatorCb.SelectedIndex = OperatorCb.FindString("+").ToString

End Sub

End Class