Response to Felix:

Hi Felix,

I chose to look at your DisplayChoice sub procedure. Your code seems to work smoothly for all four scenarios: (Y,Y), (Y,N), (N,Y), and (N,N). Your indentation for the calculating part was not aligned, so I fixed that. I also outdented your non-comment lines once since the comments indicated a new section. Next, I changed your variable names, shortening strHaveJob to strJob and replacing strSchool with strDegree. I felt that strDegree better described the variable. Lastly, I added UCase into your If clauses to account for lowercase letters being entered. Here is the code:

Public Sub DisplayChoice()

'declare variables

Dim strJob As String, strDegree As String

'assign address to values with user input

strJob = InputBox(Prompt:="Do you want to work with computers? (Y/N)", Title:="Choice")

strDegree = InputBox(Prompt:="Do you have a bachelor's degree? (Y/N)", Title:="Degree")

'processing

If UCase(strJob) = "Y" Then

If UCase(strDegree) = "N" Then

MsgBox Prompt:="Your salary will be $49,900."

Else

MsgBox Prompt:="Your salary will be $89,900."

End If

Else

MsgBox Prompt:="You need to get a job."

End If

End Sub

Response to John:

Hi John,

I chose to look at your getNestIfInElse sub procedure. Your code was pretty straightforward and I appreciated the comments. First, I added indentation to clarify which clauses were nested and to make the code more readable. Hopefully, I indented in the right places. Second, I altered your nested clauses. I changed the nested if clause to cover 0 <= intEmp < 100. Otherwise, the nested else would not have a range to cover and thus never be used. Now, the nested else covers the negative integer range and tells the user that an invalid employee ID number was entered. Here is the code:

Public Sub getNestIfInElse()

'declare variables

Dim intEmp As Integer

'assign values

intEmp = InputBox(Prompt:="Enter your employee ID number:", Title:="Employee ID")

'process and output data

If intEmp >= 100 Then

MsgBox Prompt:="You are a full-time employee. You are eligible for overtime pay."

Else

If 0 <= intEmp And intEmp < 100 Then

MsgBox Prompt:="You are a part-time employee. You are not eligible for overtime pay."

Else

MsgBox Prompt:="You entered an invalid ID number."

End If

End If

End Sub

Post:

My examples revolve around financial mathematics. They calculate the future time value of money (FV), using either simple or compound interest, with the formulas FV = PV(1+*it*) and FV = PV(1+*i*)^*t*, respectively. Note that *i*, the interest rate, should be less than one and that values greater than 1 are subsequently reduced by 1, not divided by 100 as they were in the lecture examples. Both of my examples use both a nested “If in Then” clause and a nested “If in Else” clause. The InterestCalc sub procedure begins its initial If clause by checking the type of interest rate while the CalcInterest sub procedure begins its initial If clause by checking whether or not the interest rate is greater than or equal to 1. Here is the code:

InterestCalc():

Public Sub InterestCalc()

'declare variables

Dim curPV As Currency, curFV As Currency, intTime As Integer, \_

strRateType As String, sngRate As Single

'assign values

curPV = InputBox(Prompt:="Enter PV:")

intTime = InputBox(Prompt:="Enter the duration of time (years):", \_

Default:=5)

strRateType = InputBox(Prompt:="Enter the type of interest rate " \_

& "(Simple/Compound)", Default:="Simple")

sngRate = InputBox(Prompt:="Enter the interest rate:", Title:="Rate", \_

Default:=Format(Expression:=sngRate, Format:="Fixed"))

'display labels & values

Range("A1").Value = "PV"

Range("A2").Value = "t, in years"

Range("A3").Value = "i"

Range("A4").Value = "Simple/Compound"

Range("A5").Value = "FV"

Range("B1").Value = curPV

Range("B2").Value = intTime

If sngRate >= 1 Then

Range("B3").Value = sngRate - 1

Else

Range("B3").Value = sngRate

End If

Range("B4").Value = strRateType

'processing

If UCase(String:=strRateType) = "COMPOUND" Then

If sngRate >= 1 Then

curFV = curPV \* sngRate ^ intTime

Else

curFV = curPV \* (1 + sngRate) ^ intTime

End If

Else

If sngRate >= 1 Then

curFV = curPV \* (1 + (sngRate - 1) \* intTime)

Else

curFV = curPV \* (1 + sngRate \* intTime)

End If

End If

'output

Range("B5").Value = curFV

End Sub

CalcInterest():

Public Sub CalcInterest()

'declare variables

Dim curPV As Currency, curFV As Currency, intTime As Integer, \_

strRateType As String, sngRate As Single

'assign values

curPV = InputBox(Prompt:="Enter PV:")

intTime = InputBox(Prompt:="Enter the duration of time (years):", \_

Default:=5)

strRateType = InputBox(Prompt:="Enter the type of interest rate " \_

& "(Simple/Compound)", Default:="Simple")

sngRate = InputBox(Prompt:="Enter the interest rate:", Title:="Rate", \_

Default:=Format(Expression:=sngRate, Format:="Fixed"))

'display labels & values

Range("A1").Value = "PV"

Range("A2").Value = "t, in years"

Range("A3").Value = "i"

Range("A4").Value = "Simple/Compound"

Range("A5").Value = "FV"

Range("B1").Value = curPV

Range("B2").Value = intTime

Range("B4").Value = strRateType

'processing & display sngRate

If sngRate >= 1 Then

Range("B3").Value = sngRate - 1

If UCase(String:=strRateType) = "COMPOUND" Then

curFV = curPV \* sngRate ^ intTime

Else

curFV = curPV \* (1 + (sngRate - 1) \* intTime)

End If

Else

Range("B3").Value = sngRate

If UCase(String:=strRateType) = "COMPOUND" Then

curFV = curPV \* (1 + sngRate) ^ intTime

Else

curFV = curPV \* (1 + sngRate \* intTime)

End If

End If

'Output

Range("B5").Value = curFV

End Sub