Handout 4
We will continue to work on VB programming principles. To illustrate the concepts of structured programming, we will use a loan analysis task. Assume the loan is repaid in equal monthly payments and interest is compounded monthly. The program should request the amount (principal) of the loan, the annual rate of interest, and the number of years over which the loan is to be repaid. The four options provided by command buttons are;

1. Calculate the monthly payment. The formula for the monthly payment is

\[
payment = \frac{p \times r}{1 - (1 + r)^{-n}} \quad \text{formula 1}
\]

where \( p \) is the principal of the loan, \( r \) is the monthly interest rate (annual rate divided by 12) given as a number between 0 (for 0 percent) and 1 (for 10 percent), and \( n \) is the number of months over which the loan is to be repaid. Because a payment computed here can include fractions of a cent, the value should be rounded up to the next cent. This corrected payment can be computed using the formula

\[
correct \, payment = \frac{-\text{Int}(-100 \times payment)}{100} \quad \text{formula 2}
\]

2. Display an amortization schedule, that is, a table showing the balance on the loan at the end of each month for any year over the duration of the loan. Also show how much of each monthly payment goes toward interest and how much is used to repay the principal. Finally, display the total interest paid over the duration of the loan. The balance for the successive months are calculated using the formula

\[
balance = (1 + r) \times b - m \quad \text{formula 3}
\]

where \( r \) is the monthly interest rate, \( b \) is the balance for the preceding month (amount of loan left to be paid), and \( m \) is the monthly payment.

3. Show the effect of changes in the interest rate. Display a table giving the monthly payment for each interest rate from 1 percent below to 1 percent above the specified annual rate in steps of one-eighth of a percent.

4. Quit.

Program Design
For each of the preceding tasks, the program must first look at the text boxes to read and get the loan particulars. Then the processing can begin. Therefore, we can state these four tasks to be;

1. Input the principal, interest, and duration of loan
2. Calculate the monthly payment
3. Calculate the amortization schedule
4. Display the effects of interest rate changes
5. Quit

The first task is an input operation and task 2 applies the first formula to the data. Tasks 3 and 4 need to be broken down further to simplify.

Calculate Amortization Schedule
The task involves simulating the loan month by month. First, the monthly payment is computed. Then, for each month, the new balance must be computed together with a decomposition of the monthly payment into the amount paid for interest and the amount going toward repaying the principal. The subtasks are;

3.1. Calculate the monthly payment
3.2. Calculate new balance
3.3. Calculate amount of monthly payment for interest
3.4. Calculate amount of monthly payment for principal

Display the Effects of Interest Changes
A table is needed to show the effects of changes in the interest rate on the size of the monthly payments. First, the interest rate is reduced by one percentage point and the new monthly payment is computed. Then the interest rate is increased by regular increments until it reaches one percentage point above the original rate, with new monthly payment amounts computed for each intermediate interest rate. The subtasks are:

4.1. Reduce the interest rate by one percentage point.
4.2. Calculate the monthly payment
4.3. Increase the interest rate by 1/8 percent

The tasks involved in our program can now be shown using the following function hierarchy.

![Function Hierarchy Diagram]

Figure 0. Function Hierarchy

The User Interface
The enclosed files contain the Visual Basic project and form files. A copy of the interface produced by the system is also shown here. The files are LoanAnalysisTemp.vbp and frmLoanTemp.frm. The form controls and associated procedures are available in the project files. Please note that we have 3 labels, 3 text boxes, four command buttons, and one picture box in the form. The picture box is used for displaying the program results.
Psuedocode for the Program

Presentation Layer

Calculate Monthly Payment Button – cmdPayment_Click()
  Input Loan Data – GetFormInput()
  Display Monthly Payment – ShowPayment()

Display Interest Rate Change Table – cmdRateTable_Click()
  Input Loan Data – GetFormInput()
  Display interest Rate change Effects – ShowInterestChanges()

Display Amortization Schedule – cmdAmort_Click()
  Input Loan Data – GetFormInput()
  Display Amortization schedules – ShowAmortSchedule()

Quit the program – cmdQuit_Click()
  End the program.

Input Loan Data – GetFormInput()
  Reads the text boxes and assigns their values to form level variables.

The codes for the procedures are given below.

Option Explicit

Dim principal As Single 'Form level variables
Dim yearlyRate As Single 'to hold the input values of
Dim numMonths As Integer 'loan, interest, and duration.
Private Sub cmdAmort_Click()
    GetFormInput
    ShowAmortSched principal, yearlyRate, numMonths, picDisp
End Sub

Private Sub cmdPayment_Click()
    GetFormInput
    ShowPayment principal, yearlyRate, numMonths, picDisp
End Sub

Private Sub cmdQuit_Click()
    End
End Sub

Private Sub cmdRateTable_Click()
    GetFormInput
    ShowInterestChanges principal, yearlyRate, numMonths, picDisp
End Sub

Private Sub GetFormInput()

    Dim percentageRate As Single
    Dim numYears As Integer

    principal = Val(txtAmt.Text)
    percentageRate = Val(txtApr.Text)
    yearlyRate = percentageRate / 100
    numYears = Val(txtYrs.Text)
    numMonths = numYears * 12

End Sub

**Application Layer**

Here we describe the logic for the subprograms of ShowPayment(), ShowInterestChanges(), and ShowAmortSchedule() which are invoked in the presentation layer. We use a VB module to maintain all of the application logic. The procedures are called with the arguments of principal, interest rate, number of months, and the display picture box. The values for principal, interest rate, and number of
months are set by the presentation layer procedure GetFormInput(). The picture box is a form level control object. The code for ShowPayment () is shown below.

```vbs
Public Sub ShowPayment(ByVal prin As Single, ByVal yrlyRate As Single, ByVal numMnths As Integer, picBox As PictureBox)

Dim mRate As Single
Dim prn As String
Dim apr As String
Dim yrs As String
Dim pay As Single
Dim pmt As String

mRate = yrlyRate / 12
prn = Format(prin, "Currency")
apr = Format(yrlyRate * 100, ".00")
yrs = Format(numMnths / 12, ".0")

pay = Payment(prin, mRate, numMnths)
pmt = Format(pay, "Currency")

'Display calls
picBox.Cls
picBox.Print " the monthly payment for a " & prn & " loan at " & apr
picBox.Print "% annual rate of interest for ";
picBox.Print yrs & " years is " & pmt

End Sub
```

The code for the Payment function is shown below. Format function is a standard VB function and hence no code is given.

```vbs
Private Function Payment(ByVal prin As Single, ByVal mRate As Single, ByVal numMnths As Integer)

Dim payEst As Single

If numMnths = 0 Then
```
payEst = prin

ElseIf mRate = 0 Then
    payEst = prin / numMnths
Else
    payEst = prin * mRate / (1 - (1 + mRate) ^ (-numMnths))
End If

Payment = -Int(-payEst * 100) / 100 'round to nearest cent

End Function

The code for ShowInterestChanges() is shown below. Note that the procedure also calls the Payment() function.

Public Sub ShowInterestChanges(ByVal prin As Single, ByVal yrlyRate As Single, ByVal numMnths As Integer, picBox As PictureBox)
Dim newRate As Single, mRate As Single, pmt As Single
Dim pymnt As String

picBox.Cls
picBox.Print , "Annual"
picBox.Print , "Interest rate", "Monthly Payment"

newRate = yrlyRate - 0.01

Do
    mRate = newRate / 12
    pmt = Payment(prin, mRate, numMnths)
    pymnt = Format(pmt, "Currency")
    picBox.Print , Format(newRate * 100, ".000") + ",", pymnt
    newRate = newRate + 0.01
Loop Until newRate > yrlyRate + 0.01

End Sub

The ShowAmortSched() procedure is the most complex. It not only has to call the Payment() function but also has to call the Balance() function to compute the remaining balance. The code appears below.

Public Sub ShowAmortSched(ByVal prin As Single, ByVal yrlyRate As Single, }
ByVal numMnths As Integer, _
picBox As PictureBox)

Dim msg As String
Dim startMonth As Integer
Dim monthNum As Integer
Dim loanYears As Integer
Dim mRate As Single
Dim monthlyPayment As Single
Dim totalInterest As Single
Dim interestPaid As Single
Dim yearInterest As Single
Dim oldBalance As Single
Dim newBalance As Single
Dim principalPaid As Single
Dim reducPrin As Single

msg = "Please enter year (1-" & Str(numMnths / 12) & ") for which amortization is to be shown:"
startMonth = 12 * Val(InputBox(msg)) - 11

picBox.Cls
picBox.Print "", "Amount Paid",
picBox.Print "Amount Paid", "Balance at"
picBox.Print "Month", "for Principal",
picBox.Print "for Interest", "End of Month"

mRate = yrlyRate / 12
monthlyPayment = Payment(prin, mRate, numMnths)

totalInterest = 0
yearInterest = 0
oldBalance = prin

For monthNum = 1 To numMnths
    newBalance = Balance(monthlyPayment, oldBalance, mRate)
    principalPaid = oldBalance - newBalance
    interestPaid = monthlyPayment - principalPaid
```
totalInterest = totalInterest + interestPaid

If monthNum >= startMonth And monthNum <= startMonth + 11 Then
    picBox.Print Tab(2); Format(monthNum, "#")
    picBox.Print Format(principalPaid, "Currency")
    picBox.Print Format(interestPaid, "Currency")
    picBox.Print Format(newBalance, "Currency")
    yearInterest = yearInterest + interestPaid
End If
oldBalance = newBalance
Next monthNum

reducPrin = 12 * monthlyPayment - yearInterest
loanYears = Str(numMnths / 12)

picBox.Print
picBox.Print "reduction in principal",
picBox.Print Format(reducPrin, "Currency")
picBox.Print "interest paid",
picBox.Print Format(yearInterest, "Currency")
picBox.Print "Total interest over" & loanYears & " years",
picBox.Print Format(totalInterest, "Currency")

End Sub```