Please use your own paper to answer the questions. Turn in your answers as well as the test document. Be sure that your name is on your answer pages.

Question point values are shown in parentheses.

1. (20) Assume that you have a text box named txtTest. Write the appropriate procedure(s) so that:

   a. The user is limited to entering only digits, a decimal point, a minus sign, or backspaces. Any other characters should be rejected and the computer should beep. Do not worry about the location or number of decimal points or minus signs.

   b. The value that the user enters is automatically changed to currency format when the user selects another control.

2. (16) Assume that the following statements have been executed.

```
Dim A As Integer
Dim B As Single
Dim C As Currency
Dim D As Single
Dim E As String
Dim F As String

A = 10
B = 15
C = 5.5
D = 2
E = "Able"
F = "Mable"
```

   Evaluate each expression below and indicate its value.

   a. B / D + C * 2

   b. A + B * D ^ 2

   c. B + -D - A / A

   d. Left$(E, 2) & Right$(F, 2) & Mid$(F, 2, 4)

3. (12) A graduate school of business has some rules that is uses to determine how applicants should be evaluated for admission. One factor it uses is the applicant's score on the Graduate Management Aptitude Test (GMAT). Another factor is the applicant's undergraduate GPA.

   The specific rules are summarized below.

		GMAT		
		< 600	600 - 700	> 700
GPA	< 3.0	Reject	Reject	Consider
	3.0 - 3.5	Reject	Consider	Accept
	> 3.5	Consider	Accept	Accept

   Assume that variables named GMAT and GPA have been defined and contain valid values. Using these variables, write the appropriate VB contingency-handling statement(s) to set a string variable named Decision equal to "Reject", "Consider", or "Accept" as appropriate.

4. (16) How are the concepts of variable scope and the use of arguments and parameters in procedures similar? How are they different?

5. (18) You are given the following code:

```
Option Explicit

Dim A As Integer    'This is in the form's general declaration area
```

```
Sub Command1_Click ()
Dim A As Integer
Dim B As Integer

A = 10
B = 20

FirstCheck A, B
MsgBox Str$(A) & " " & Str$(B)

SecondCheck A, B
MsgBox Str$(A) & " " & Str$(B)

ThirdCheck A, B
MsgBox Str$(A) & " " & Str$(B)

End Sub
```

```
Sub FirstCheck (X As Integer, ByVal Y As Integer)

A = X + Y
X = Y * 2
Y = 30

End Sub
```

```
Sub SecondCheck (K As Integer, M As Integer)
Dim R As Integer

R = K + A
M = R

End Sub
```

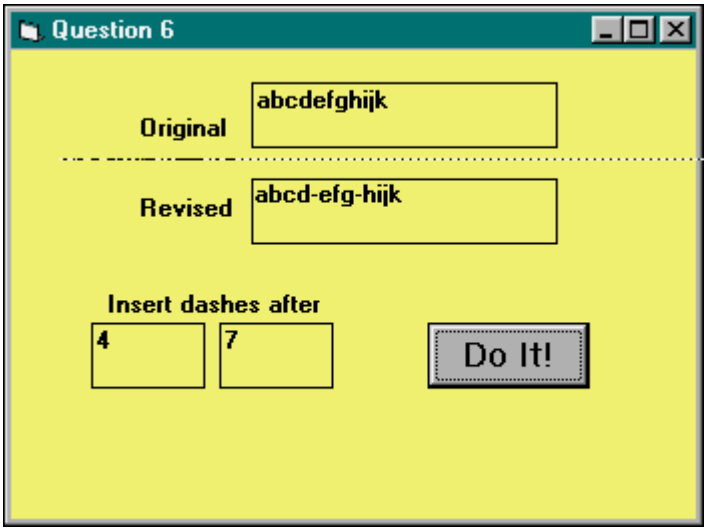
```
Sub ThirdCheck (B As Integer, A As Integer)

SecondCheck A, B
FirstCheck B, A

End Sub
```

What output is generated by each of the three MsgBox statements?

6. (18) Write a general function that inserts two dashes into a string. The user should be able to specify the locations where the dashes are to be inserted. The screen shot below shows the project while it is executing.



In addition to the function inserting the dashes, it should also check to make sure that the first dash location is less than or equal to the second dash location. It should also make sure that the dash locations are not beyond the end of the original string. If either of these conditions are violated, the function should display a message box indicating the nature of the problem and it should return the word "Error" instead of a revised string.

Just write the function, not the entire project (as shown above).

Answer Key

1. Two event procedures are needed - a KeyPress event and a LostFocus event.

```
Sub txtTest_KeyPress ( KeyAscii As Integer)

Select Case KeyAscii
Case Asc("0") To Asc("9"), Asc("."), Asc("-"), 8
    'OK - do nothing
Case Else
    Beep
    KeyAscii = 0
End Select

End Sub

Sub txtTest_LostFocus ()

txtTest.Text = Format$(txtTest.Text, "Currency")

End Sub
```

2. a. 18.5
   b. 70
   c. 12
   d. Ableable

3. The following is the solution using Select Case. An If…Then…Else solution or a combination of both is also acceptable.

```
Select Case GMAT
Case Is < 600
    Select Case GPA
    Case Is < 3
        Decision = "Reject"
    Case 3 To 3.5
        Decision = "Reject"
    Case Is > 3.5
        Decision = "Consider"
    End Select
Case 600 To 700
    Select Case GPA
    Case Is < 3
        Decision = "Reject"
    Case 3 To 3.5
        Decision = "Consider"
    Case Is > 3.5
        Decision = "Accept"
    End Select
Case Is > 700
    Select Case GPA
    Case Is < 3
```

```
            Decision = "Consider"
        Case 3 To 3.5
            Decision = "Accept"
        Case Is > 3.5
            Decision = "Accept"
        End Select
    End Select
```

4. They are similar in that they both facilitate the sharing of variables between procedures. The difference between them is in the amount of control one can exert on this sharing. Variable scope allows sharing but does not limit the sharing very effectively. For example, a module-level variable is visible to all procedures in the module. An argument is only visible to the specific procedure that it was passed to. In addition, parameters can be restricted to read-only access by using the ByVal option. All variables shared by module or global scope are read / write, i.e., any procedure that has access to them can both read and write to them.

5. 40 20

   40 70

   140 70

6.

```
Function InsertDashes ( ByVal FirstLoc As Integer, ByVal SecondLoc
      As Integer, ByVal Word As String) As String
Dim Temp As String

If FirstLoc > SecondLoc Then
    MsgBox "First location must come before second."
    InsertDashes = "Error"
    Exit Function
End If

If FirstLoc > Len(Word) Or SecondLoc > Len(Word) Then
    MsgBox "Locations must be within the original string."
    InsertDashes = "Error"
    Exit Function
End If

Temp = Left$(Word, FirstLoc) & "-"
Temp = Temp & Mid$(Word, FirstLoc + 1, SecondLoc - FirstLoc) & "-"
Temp = Temp & Right$(Word, Len(Word) - SecondLoc)

InsertDashes = Temp

End Function
```