Program Example 22  Message & Input Box

Objectives

**Program Purpose**
- Display 1 of 3 possible messages depending on the number selected using a vertical scroll bar
- Gain the user’s name through an Input Box

**Learning Goals**
- Use of Input Box and Message Box
- Use of Constants
- SELECT..CASE statement
- Use of Trim and Val Functions
- Use of WHILE..WEND

Design Notes

This project is a simple exercise, demonstrating the use of Input and Message Boxes. One of three possible messages is displayed depending on the number selected. Constants are used to organise the desired messages more effectively. Having all the messages in the one section enables them to be easily modified and expanded if necessary, without accidentally changing any code.

In this example, the selection of a low number is rewarded by a ‘good’ message. This is entirely arbitrary.

Interface

Create the interface as shown.
Use 1 form, 1 label, 1 vertical scroll bar and 2 command buttons.

![Message & Input Box Interface](image)

Names of Objects

<table>
<thead>
<tr>
<th>Type of Object</th>
<th>Number</th>
<th>Names of Objects</th>
<th>Simple Initial Properties of Objects</th>
</tr>
</thead>
</table>
| Form                   | 1      | Form1                  | Caption - "MsgBox & Input Box 1"  
Startup Position - 2 (Center Screen)                                                                 |
| Vertical Scroll Bars   | 1      | vsbNum                 | Min - 8  
Max - 40  
Value - 8                                                                                              |
| Labels                 | 1      | Label1                 | Font - Bold, 14  
Backstyle - opaque  
Borderstyle - Fixed Single  
Caption - ""                                                                                           |
| Command Buttons        | 2      | cmdMessage, cmdGetName | Font - Bold, 12  
Captions - as per Interface                                                                          |
Code

**GENERAL SECTION**

Const MsgHigh = "Well Done! You're a great player"
Const MsgMed = "OK, You're a boring player"
Const MsgBad = "Your playing is very bad"

**EVENTS**

*Private Sub cmdGetName_Click()*

Dim InText As String
InText = InputBox("Enter your Name", "Getting Your Name", "Anthony")
While Trim(InText) = ""
    InText = InputBox("Enter Your Name - No spaces, please", "Getting Your Name", "Anthony")
Wend
MsgBox "Your name  = " & InText, vbOKCancel + vbInformation, "InputBox"

*End Sub*

*Private Sub cmdMessage_Click()*

Select Case Val(Label1.Caption)
    Case 8 To 15
        MsgBox MsgHigh, vbYesNo + vbInformation, "Program Help File"
    Case 16 To 24
        MsgBox MsgMed, vbYesNoCancel + vbCritical
    Case Else
        MsgBox MsgBad, vbDefaultButton1 + vbExclamation + vbYesNoCancel
End Select

*End Sub*

*Private Sub Form_Load()*

Label1.Caption = vsbNum.Value

*End Sub*

*Private Sub vsbNum_Change()*

Label1.Caption = vsbNum.Value

*End Sub*

**Consolidation and Extension**

1. Add 2 more message constants for a greater range of numbers. Modify the properties of the scroll bar to accommodate your choice.
   - e.g. Const MsgVeryBad = “You should give up this game”
   - Const MsgExcellent = “You are awesome!”

2. Output the user’s name to the form caption rather than to a message box.

3. Display the messages in the label rather than in a message box.

4. What is the function of the vbDefaultButton1 and vbDefaultButton2 constants?

5. Create a separate project that asks for the user’s age and displays a range of messages accordingly. Use the IsNumeric function to check that the user is typing in a number. Validate the range of the number entered.

*Note: The variable used for input cannot be declared as an integer; otherwise an error will occur when a user types in non-numeric input. A string variable is used to collect the input. A check is then made to see if the input is numeric.*