Learning Objectives:
1. To create a UserForm that interacts with the user.
2. To create a procedure the opens the UserForm
3. To learn how to assign code to objects in a UserForm

UserForms

- UserForms are custom dialog boxes that the programmer can prepare to perform specific functions.
- A UserForm can contain objects, such as command buttons, option buttons, check boxes, labels, and textboxes. A UserForm has to be “shown” to be functional. A short procedure in a module is used to open a UserForm.
- UserForms have their own code window. All code associated with a UserForm is placed in this code window.
- UserForms are created in the Visual Basic Editor (VBE). Enter the VBE and right-click on one of the objects in the project manager. Select Insert – UserForm.

You can change the name of the UserForm by clicking in the box next to (Name) in the properties window for UserForm1. UserForm names usually begin with the prefix “frm” to indicate its object type in the name. We will leave the name of the UserForm as UserForm1. The caption can be changed to read “Formatted Text”. This caption will be displayed in the titlebar of the UserForm.

Objects are added to the UserForm from the Toolbox, which is shown below.
We are going to create a custom userform that will place formatted text into the active cell of a worksheet. The intent is to have a normal text string and a text string that is subscripted. One of the options is to have the normal text string be bold or italic or symbol font. There is an optional equal sign after the text.

To add objects to the UserForm, click on the object's button in the toolbox and click in the UserForm and drag to get the size you need. It is just like making a drawing object in Word. Reproduce the objects shown below:

- Click on each object once it is created and make sure the name in the properties window matches the name given above exactly. If a name doesn't match, correct it in the properties window.
- Click on OptionButton1 and change its caption to "Symbol Font" in the properties window.
- Click on CheckBox1 and change its caption to "Symbol Font" in the properties window. Also change its Value to True in the properties window.
- Click on CheckBox2 and change its caption to "Bold" in the properties window.
- Click on CheckBox3 and change its caption to "Italic" in the properties window.
- Click on CommandButton1 and in the properties window, change its caption to "Place Text"
- Click on CommandButton2 and in the properties window, change its caption to "Cancel". Your UserForm should look like the figure above.

Now, we add the code for the Cancel button. Double-clicking on the Cancel button opens the code window for the UserForm. Text is automatically generated which looks like:

```vba
Private Sub CommandButton2_Click ()
    UserForm1.Hide
End Sub
```

The "_Click" is the event that is required to execute this code. In this case, the code executes when the user clicks on the command button. VBA objects have a variety of events that can be used to execute code.

- Between the two lines of code, add the line: `UserForm1.Hide` This will hide the UserForm when the user picks the cancel button.
- Double-click on the UserForm1 in the project manager to switch back to the UserForm.
When the UserForm is opened, we plan for the user to type the normal text in TextBox1 and the subscript text in TextBox2. Our project needs to get the text from these text boxes and join them together. Then, we need to format the text according to the option buttons and check boxes. The process for doing this task is:

1. Create a variables TheText, Length1, Length2
   (TheText is the text string to be formatted. Length1 is the length of the text in TextBox1, and Length2 is the length of text in TextBox2)

2. If the value of CheckBox1 is true then the value of TheText is TextBox1 & Textbox2 & " = "
   Otherwise, it is just TextBox1 & Textbox2.

3. Place TheText into the activecell

4. If OptionButton1 is true then the name of the font for the first group of characters (Length1 long) is Symbol.

5. If CheckBox2 is True then the font for the first group of characters (Length1 long) is Bold.

6. If CheckBox3 is True then the font for the first group of characters (Length1 long) is Italic.

7. The font for the second group of characters (starting with Length1 + 1 for Length2 characters) is subscript.

8. Close the UserForm

Double-click on the “Place Text” button and add the following code:

```vba
Private Sub CommandButton1_Click()
    Dim TheText As String
    Dim Length1, Length2 As Integer

    Length1 = Len(TextBox1.Text) 'finds length of first text string
    Length2 = Len(TextBox2.Text) 'finds length of second text string

    If CheckBox1.Value = True Then
        TheText = TextBox1.Text + TextBox2.Text & " = " 'adds equal sign if checked
    Else
        TheText = TextBox1.Text + TextBox2.Text 'no equal sign if unchecked
    End If

    ActiveCell.Value = TheText 'places text in activecell

    If OptionButton1.Value = True Then
        ActiveCell.Characters(1, Length1).Font.Name = "symbol" 'symbol font if selected
    End If

    If CheckBox2.Value = True Then
        ActiveCell.Characters(1, Length1).Font.Bold = True 'bold if checked
    End If

    If CheckBox3.Value = True Then
        ActiveCell.Characters(1, Length1).Font.Italic = True 'italic if checked
    End If

    ActiveCell.Characters(Length1 + 1, Length2).Font.Subscript = True 'subscript

    UserForm1.Hide 'turns off the UserForm
End Sub
```
Next, we need to generate a sub procedure that will open the UserForm for the user to use. Right-click in the project manager and select **Insert – Module.** In the module, create the following procedure:

```vba
Sub ShowUserForm1 ()
    UserForm1.Show
End Sub
```

This procedure can be accessed through the run options of the macro menu in Excel. It can also be assigned to a custom button in Excel.

Exit the VBE and try it out… Pick the run button from the Visual Basic toolbar and pick on the “ShowUserForm1” and pick OK. I would create a custom button and assign the ShowUserForm1 procedure to the button for one click action!