**Computer Programming 1
Essential Standard 7.00 Apply Advanced Logic
7.01 Apply sub procedures/methods and user defined functions.**

In all programs, put your name, the assignment name and the date in comments at the top.

**Walk-Through Exercises**

1. Create a project called TwinkleTwinkle. Add the controls shown below with appropriate naming.

	1. In the Code Editor window.
		1. Type the following to create your sub procedure.

		 Private Sub displayPoem()
		2. When you hit the enter key, End Sub will appear.
		3. Between the Private Sub and End Sub, type the following.

		 Me.lblDisplayPoem.Text = "Twinkle Twinkle Little Star" &
		 vbCrLf & "How I Wonder What You Are" & vbCrLf & "Up Above
		 The World So Bright" & vbCrLf & "Like a Diamond In The
		 Night"
		4. Create a button click event.
			* Inside the button click, add the code to “call” your displayPoem sub procedure.

			displayPoem()
		5. Execute your code.
2. Create a project called Multiples. Add the controls shown below with appropriate naming.

	1. In the Code Editor window.
		1. Type the following to create your sub procedure. You will have a value parameter that will hold the number from the textbox that will be “sent” in the “call”

		 Private Sub showMultiples (ByVal intNum As Integer)
		2. When you hit the enter key, End Sub will appear.
		3. Enter the code that will loop to display the results of multiplying the number by each of the numbers 1 through 10.

		 ‘loops through 1 to 10 to create multiple
		 For intTime As Integer = 1 To 10

Me.lblShowMultiples.Text &= intNum & " x " & intTime & " = " & intNum \* intTime & vbCrLf
’adds text to label with return

 Next intTime

* + 1. Create a button click event.
			- Inside the button click, add the code to get the input from the user (using a Try Catch) and calling the sub procedure.

			Dim intNumEntered As Integer ‘input from user
			Try
			 intNumEntered = Convert.ToInt16(Me.txtNum.Text)
			 showMultiples(intNumEntered) ‘calls method & send input
			Catch ex As Exception
			 MessageBox.Show("Enter a numeric value")
			 ‘error message
			End Try.
		2. Execute your code
1. Create a project called TimesTen. Add the controls shown below with appropriate naming.

	1. In the code editor window.
		1. Type the following to create your sub procedure. You will have a value parameter that will hold the number from the textbox that will be “sent” in the “call” as well as a reference parameter that will sends its “address” so that the value can be “passed” back.

		Private Sub MultiplyByTen(ByVal intNum As Integer, ByRef intAnswer As Integer)
		2. When you hit the enter key, End Sub will appear.
		3. Add the code in the sub that will calculate the answer variable by multiplying the number from the textbox by 10

		 'sets the answer in intAnswer's location
		 intAnswer = intNum \* 10
		4. Create a button click event.
		5. Inside the button click, add the code to get the input from the user (using a Try Catch) and calling the sub procedure.

		 Try

intNumEntered = Convert.ToInt16(Me.txtNumEntered.Text)

'sends inNumEntered value and intAnswer location

 multiplyByTen(intNumEntered, intAnswer)

 'displays answer

 Me.lblAnswer.Text = intNumEntered & " x 10 = " & intAnswer

 Catch ex As Exception

 MessageBox.Show("Enter a numeric value")

 End Try

* 1. Execute your code
1. Create a project called Signature. Add the controls shown below with appropriate naming.

	1. In the code editor window.
	2. Type the following to create your function. You will have a value parameter that will hold the name from the textbox that will be “sent” in the “call”. Note that you are using Function instead of Sub as the keyword to create the function and there is a return data type at the end.

	Function Signature(ByVal strName As String) As String

Dim strToReturn As String

 ‘create the string to return

 strToReturn = "Create By " & strName

 Return strToReturn ‘sends the string back.

End Function

* 1. Create a button click event.
	2. Inside the button click, add the code to get the input from the user and call the function. Remember a function returns a value, therefore it should be on the right side of an assignment statement.

	 Dim strFName As String
	 strFName = Me.txtFName.Text
	 Me.lblDisplay.Text = Signature(strFName)
	3. Execute your code.