MIS 216
FALL 2015
PROJECT 5

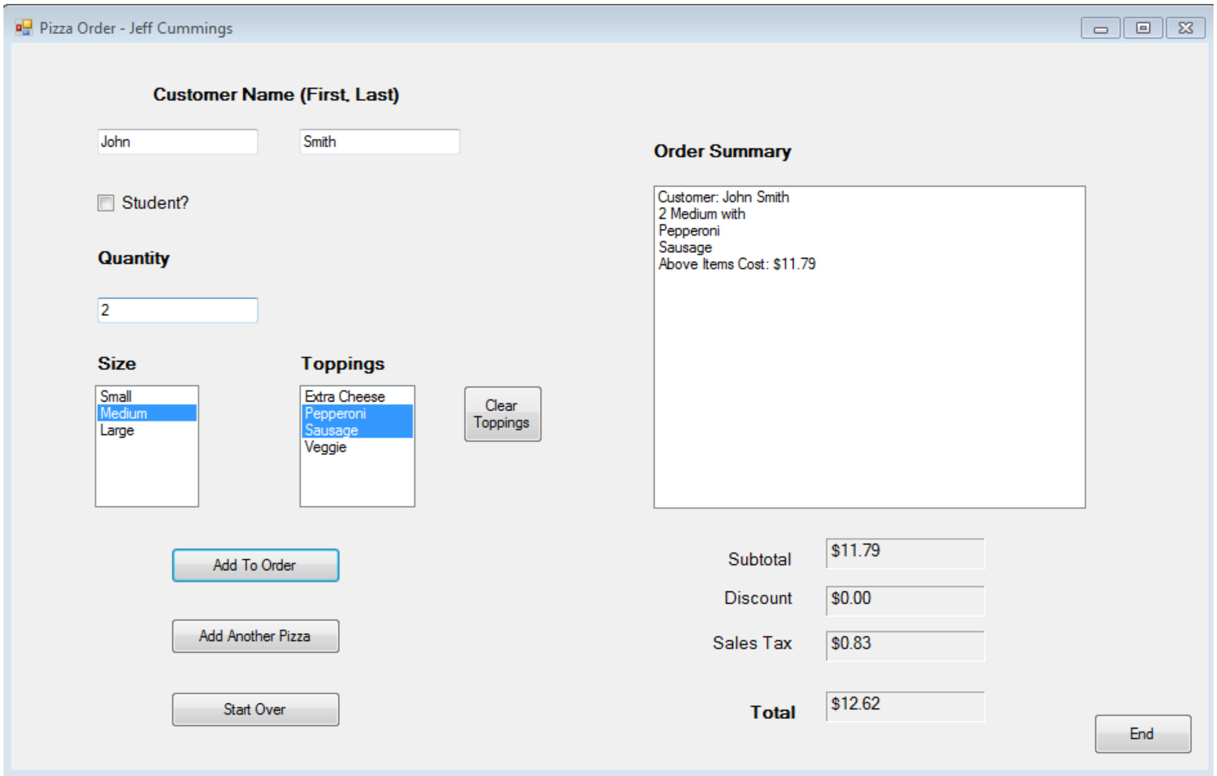
Putting it all together

1. Start a new project on your desktop, name the project folder (yourinitialsProject5) as in tnjProject5.
   a. FILE – NEW PROJECT
   b. Change the Path to Desktop
   c. Change the name, solution name to yourIntialsProject5
   d. Add a folder name after \desktop\ in the location as in yourInitialsProject5

2. It will build a starting form for you, change the text property to:
   Pizza Order – Your Name

3. This project will reinforce all of your skills and practices we learned this semester.

4. **Remember the basics for all pages/calculations**
   a. Add a comment to the beginning of all new sub's or functions
   b. Test for valid input (number, blanks, selections)
   c. Move the input text boxes to variables using the proper CType
   d. Do any necessary calculations
   e. Move to the output areas, and format as necessary



Customer Name (First, Last): John Smith

□ Student?

Quantity: 2

Size: Medium

Toppings: Pepperoni, Sausage

Order Summary:

```
Customer: John Smith
2 Medium with
Pepperoni
Sausage
Above Items Cost: $11.79
```

Subtotal	$11.79
Discount	$0.00
Sales Tax	$0.83
Total	$12.62

5. First create the form shown, please use proper naming on all objects (txt, lbl, chk, lst, btn). To help you there are,
    a. 3 text boxes
    b. 1 check boxes
    c. 3 list boxes (**hint:** to add the sizes and toppings use the "Items" property)
    d. 5 buttons
    e. Many, many labels

6. Declare the following 'global variables' to be use by all subs and functions (place these after the top Public Class xxx line).
    a. sngSubTotal
    b. sngDiscount
    c. sngTaxes
    d. sngTotal
    e. intQuantity
    f. blnFirstPieOrdered

<u>Add To Order Button</u>

7. Check for valid data in the text boxes and list boxes before any calculations, remember if you find an error, display the error and EXIT SUB (hint: you can use a function).
    a. To check if a 'string' has been entered you check the length of the entry as in:
        i. Me.*NameOfTextBox*.**Text.Trim.Length = 0**
    b. To check if a size of the Pizza has been selected in the list box, you verify that one has been selected as in::
        i. Me.*NameOfListBox*.**SelectedItems.Count = 0**
    c. Check to see if the quantity ordered is > 0, if not generate an error message

8. If all error checking passes, then move the value from the Quantity text box entry to the global variable previously declared.

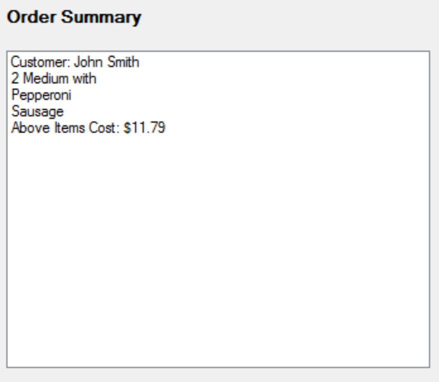
9. The first line of the summary list box should report (see figure):

    a. You will want to use an if statement to add the customer name so it will only appear once.

```
If blnFirstPieOrdered = False
  lstSummary.Items.Add("Customer: " & txtFirstName.Text & " " & _
  txtLastName.Text)

  blnFirstPieOrdered = True
End If
```

10. You then want to add to the summary list the number of pizzas of a particular size and toppings ordered, here are some examples (instructions on how to do this starts at 11):



```
Customer: John Smith
2 Medium with
Pepperoni
Sausage
Above Items Cost: $11.79
```

11. To get the size selected, the property is: *name of listbox*.SelectedItem.ToString

12. For the toppings, first check to see if something is selected using an if statement (use *ToppingsListBox*.SelectedItems.Count to see if this is = 0 meaning no toppings have been selected).

    If something is selected, you will need to list those toppings in the listbox. To get multiple toppings, first change the "**SelectionMode"** property of the toppings list box to "**MultiSimple"**.

The following loop will copy the name of all those selected to your summary list:

```
For intCtr = 0 To Me.ToppingsListBox.SelectedItems.Count - 1
   Me.SummaryListBox.Items.Add(Me.lstToppings.SelectedItems(intCtr).ToString)
Next
```

13. .Create a new function to calculate the cost with or without toppings. Nothing needs to be passed to the function but the function should return a ***single*** variable type with the cost for one of the order components. Your function should have two Select Case / End Select statements in it. Here are the rules:

    Create a variable to store your cost locally in the function (e.g. sngCost)

    a. Select Case 1: use the *SizeListBox*.Text to check for the following and then set sngCost equal to the appropriate cost of the pizza size
        i. Small – 7.99
        ii. Medium – 9.99
        iii. Large – 11.99

    b. Select Case 2: use *ToppingsListBox*.SelectedItems.Count to determine how many toppings were selected and add this to the sngCost (be sure to use += so you don't overwrite your pizza size cost)
        i. One Topping - $1.00 extra

ii. Two Toppings - $.090 each or $1.80

iii. Three or More, $.080 each topping

Thus:

A small pizza with 1 topping would be 8.99
A medium pizza with 0 toppings would be 9.99
A large pizza with 3 toppings would be 14.39

14. After the value is returned to your add button routine, on the next line of code, add the output of the function to your global variable tracking the subtotal of the order.

15. Once one size/toppings has been calculate display in the summary list box as statement: **Above items cost: 14.39** (see figure on page 1)

16. Calculate the cost/tax/etc at this point. Create a **subroutine** to calculate these values. You will need to use your global variables to calculate these values.

    Here are the rules,

    a. Subtotal – is the sum of all the subtotals as described in #14
    b. Discount – if the order is from a student (*nameofCheckBox*.**Checked** = True)
        i. Give a 10% discount on the order
    c. Sales Tax is 7% of the Subtotal
    d. Total is the sum of the subtotal and sales tax less discount

17. When all values are correct format all to two decimals and move to the proper labels.

<u>Other buttons</u>:

18. The Clear Toppings button gives the clerk the option to 'deselect' a topping if clicked in error. The command is *me.nameoflistbox*.**SelectedItems.Clear**

19. The Add Another Pizza button lets the customer add either another size or another series of toppings to the order. You should clear the quantity, size and toppings objects.

20. The Start Over, should clear ALL items (including your global variables).

Please check your application before submitting,

a) Do all subs/functions have a comment
b) Are boxes cleared property, including the label where errors are displayed
c) Does the name of the customer only appear once in the summary box
d) Have fun with the last project