**Assignment #1**:

In class, we built a new window form with 2 textboxes to gather 2 numbers and a label to display the sum of those numbers when the “Add” button is pressed. For a class exercise, I would like you to modify the program to use Doubles instead of Integers and add buttons/code to perform Subtraction, Multiplication and Division.

**Lab Assignment:**

1. Create a new form that will represent a more typical calculator with one textbox to represent the LCD display.
2. Create a series of buttons to represent the number keys. (Ex: “1”, “2”, “3”, etc.) These numbers will be appended to the LCD textbox.
3. Create a series of Operator buttons to handle Addition, Subtraction, Multiplication, and Division button. The math buttons with store the current number within the LCD, the operation itself, and clear the LCD.
4. Create an Equals button that will gather the second number in the LCD display, perform the math, then display the result in the LCD display.

BONUS: (+10 points) If you can code the math operator buttons so that it can perform math in succession like this: 98 + 2 + 5 = and get the result of 105

**Assignment #2**:

In class this week, we started talking about the process of generating a form to gather information. For our example of this, we are using a contact manager as a basis for our program/form.

You need to create a new windows program that will gather the following information from the user. Use textboxes when not specified below:

Salutation (Group Box and Radio Buttons)

First Name

Last Name

Street1

Street2

City

State(Combo Box)

Zip

Email

Home Phone

Work Phone

Cell Phone

Birthday (Date/Time Picker)

Anniversary (Date/Time Picker)

Card Worthy (Checkbox)

Contacts (ListBox)

Feedback (Label)

Create a "Clear" button to clear the First and Last Names as well as the listbox.

Create an "Add" button to add the contact to an array, then use the array to fill the listbox.

**Assignment #3**:

In class this week, we created a basic class to be used for validating some of our form data and letting the form know the results so the Window form designed can use that data to give the user feedback if there are data-entry issues.

You need to create a series of functions and implement them for validating the following:

Either the First Name or the Last Name needs to be filled in. (At least one of them filled in)

The user must choose a State, do not have a state chosen by default. Instead say something like "Please choose one" be the default when the program starts.

Zip (Exactly 5 characters in length)

Email :

At least 8 characters in length,

"@" must exist and have at least 2 characters in front of it

At least 1 period must exist and the last one should have at least 2 characters after it

there must be at least 2 characters between the "@" and the last "." (period)

Home Phone (If it is filled in, it needs at least 7 characters in length)

Work Phone (If it is filled in, it needs at least 7 characters in length)

Cell Phone (If it is filled in, it needs at least 7 characters in length)

**Assignment #4**:

In class this week, we learned about string functions and properties including, but not limited to IndexOf( ), SubStr( ), Length, IsEmpty( ), and LastIndexOf( ). We are going to use these functions to create a "Jedi Name" based on some information we will gather from the user.

I need you to gather the following information from the user:

1) First Name

2) Last Name

3) Mother's Maiden Name

4) City/Town where you were born

Next, Gather the specified pieces of the gathered data and concatenate within one string variable

1) Get the first 3 letters of the Last Name

2) Get the last 2 letters from the First Name

3) Add a hyphen "-"

4) Get the first 2 letters of the Mother's Maiden Name

5) Get the last 3 letters from the Birth City/Town

Display the results to the user

Test the results:

Using the information above, use a paper and pen to calculate your Jedi Name on paper and see if it matches...if not...troubleshoot.

BONUS (+10) – Censor out a list of bad words. Display the # of asterisks to represent the length of the bad word. You must find the bad words no matter what case they are.

**Mid-Term Exam Project**

We have been hired to develop a Pizza Order Entry System. This will occur in a series of upgrades as different phases are completed.

**Phase 1- Get the order for one pizza including the following:**

Pizza Size: Radio Buttons: Small ($7), Medium ($10), Large ($12)

Pizza Toppings: Checkboxes: (+50 cents per topping):

Pepperoni, Sausage, Meatball, Ham

Peppers, Onions, Olives, Spinach

Pineapple, BBQ Sauce, Xtra Cheese, Spit

Calculate the Total Cost and Display the pizza order in a ListBox

**Phase 2 – Get the customer’s Delivery Info**:

Gather the following data from the user:

First Name, Last Name, Street1, Street2, City, State, Phone, Email

Use a Combo/Drop-Down List for the City (Delivery in only 3 local cities, default it to a specific city)

Use a Combo/Drop-Down List for the State (Default it to a specific state.)

Have the address listed either in the ListBox from Phase 1 or another output option.

**Phase 3 – Validate Customer Info**

You need to validate the following, using re-usable functions in a library:

Either the First Name or the Last Name needs to be filled in. (At least one of them filled in)

Email :

At least 8 characters in length,

"@" must exist and have at least 2 characters in front of it

At least 1 period must exist and the last one should have at least 2 characters after it

there must be at least 2 characters between the "@" and the last "." (period)

Home Phone (If it is filled in, it needs at least 7 characters in length)

+5 Bonus ->Option to remove oldest item from the Pizza Order List Box

+5 Bonus -> Use Date/Time Picker and variables within your order to enter the current date and time or later times and dates for pre-orders (validation)

**Assignment #5**:

In class this week, we learned about writing text to a file. We also chatted about comma-delimited text used as data. Modify your contact manager to append any new contacts to one specific comma-delimited text file. (All contacts stored in one file.)

1) When the user adds a contact, concatenate all the gathered data into a comma-delimited string.

2) Open a specific text file for appending.

3) Append/Write the new data to the file.

4) Open the text file using Notepad or some other text file reader to see if the data has been saved.

5) If it is comma-delimited, try opening it with Excel.

**Assignment #6**

In class this week, we learned about reading a text file and parsing the comma-delimited data. We want to be able read the data from Assign #5 and display it within our contact manager's ListBox, showing an example of bringing up saved data.

1) Use the sample Read function As-Is, read the entire file, then display entire file string to ListBox

2) Modify the Read function to loop thru the file, reading one line at a time, and copying each line to the ListBox as separate entries.

3) Modify #2 so that we parse the comma-delimited string into separate pieces of data/variables, and print specific data (First Name, Last Name, and Email) to the ListBox

**Assignment #7**

In class this week, we learned about reading a text file and parsing the comma-delimited data into separate Arrays of data. For example, you may have an array named "FName" that will hold all the first names and an array named "LName" that will hold the last names.

Once we have all the data stored in arrays, we can look at one "record" by using the same index number for each array. So...for example FName[2] and LName[2] should be the first and last names of one specific person. This allows us to pull up one person's data up via an index (subscript) number...in this case, "2" was the index/subscript.

1) Create individual arrays to hold each person's data (FName, LName, Email, etc)

2) Adjust your loop for parsing the data so that  
 a) The arrays are loaded with the parsed data  
 b) Add the data from the current FName and LName to the ListBox  
 c) Increment the counter variable that represents the index/subscript

3) Now that all the data is stored in the arrays, we want to be able to display one person's data at a time within the textboxes of the form.

This can be done by one of the following methods:

a) Create Next and Previous buttons on the form that will either increment or decrement a counter and then fill out the form with the one "record" of data using the counter variable as the index/subscript.

b) Double-click a person's name in the ListBox, which could return the index/subscript of the person clicked. Then use that index/subscript to pull up the person's info via the arrays (FName, LName, etc.).

**Assignment #8**

Continue from the previous assignment and add the capability to update your data

Final

Basic gist of the Final Exam Project is to utilize your Midterm along with fixes recommended on the midterms. From this point you will update the project to:

* Write the data to a comma-delimited text file
* Loop through the records and Read in the comma-delimited file one line at a time
* For each record parse the data into separate data variables, then into the appropriate lists.
* Build the interface to allow scrolling through the records one at a time, displaying the current record in the form.
* BONUS (10 points) – Add the ability to update the data and save the updates in the file.

**Sample Project from Midterm with additional features for Final**:

We have been hired to develop a Pizza Order Entry System. This will occur in a series of upgrades as different phases are completed. Phases 1 thru 3 should be complete already. For the additional stuff for the Final Exam, you can look at Phase 4 and higher.

**Phase 1- Get the order for one pizza including the following:**

Pizza Size: Radio Buttons: Small ($7), Medium ($10), Large ($12)

Pizza Toppings: Checkboxes: (+50 cents per topping):

Pepperoni, Sausage, Meatball, Ham

Peppers, Onions, Olives, Spinach

Pineapple, BBQ Sauce, Xtra Cheese, Spit

Calculate the Total Cost and Display the pizza order in a ListBox

**Phase 2 – Get the customer’s Delivery Info**:

Gather the following data from the user:

First Name, Last Name, Street1, Street2, City, State, Phone, Email

Use a Combo/Drop-Down List for the City (Delivery in only 3 local cities, default it to a specific city)

Use a Combo/Drop-Down List for the State (Default it to a specific state.)

Have the address listed either in the ListBox from Phase 1 or another output option.

**Phase 3 – Validate Customer Info**:

You need to validate the following, using re-usable functions in a library:

Either the First Name or the Last Name needs to be filled in. (At least one of them filled in)

Email :

At least 8 characters in length,

"@" must exist and have at least 2 characters in front of it

At least 1 period must exist and the last one should have at least 2 characters after it

there must be at least 2 characters between the "@" and the last "." (period)

Home Phone (If it is filled in, it needs at least 7 characters in length)

**Phase 4- Write & Read from a Text file holding sales information from the order entry system**

Modify the order entry form in order that it will be used to write/read/scrollthrough the sales.

(Sales data consists of the Pizza order plus the First and Last name of the customer)

Create any other necessary labels, buttons, etc..

Write the code that uses your FileIO function library(class) in order to read/write the comma-delimited data within a file you will name “orders.csv”

**Phase 5- (10 point Bonus)**

Add the ability to Update Order data and update the text data file