Introduction to Visual Basic II

Topics

- Get familiar with the VS Express 2013 environment
- String and number variables
- Assignment statements
- Evaluating expressions
- User input: strings and numbers
- Using `FormatCurrency` function
- Using format strings in Visual Basic
- Form-level variables

Video rental store

5. Use **File Explorer** to navigate to the folder containing the final version of your program `VideoRentalStore_Part3` in your lab folder. Copy and paste this program in the same folder, and then right-click on the copy to rename it as `VideoRentalStore_Part5`. Launch VS Express and use it to open the program `VideoRentalStore_Part5`.

   Now change your `VideoRentalStore_Part5` to use the `FormatCurrency` function and *format strings* in Visual Basic to produce an itemized receipt, containing the same information as the last part, but formatted EXACTLY like the following example.

```
Customer name: Mary Smith
Rental period: 5 days

Regular videos (5 @ $2.50)   $62.50
Premium videos (3 @ $5.25)   $78.75
-----------------------------------
Subtotal                    $141.25
Tax                           $9.89
Total                       $151.14
```

   You must use a format string variable to space your output appropriately. Be sure to set the font of your output textbox to `Courier New` in order for output to display properly. An example of how to declare and use a format string can be found on the next page.

Your program should behave identically to the solution which can be run by double-clicking the file `VideoRentalStore_Part5.exe` found in the `Executables` folder inside of your lab folder.

When done, save and close program `VideoRentalStore_Part5`.

Checkpoint 5 (80/100): A successful `VideoRentalStore_Part5` program:

- produces an itemized receipt formatted EXACTLY as shown in the previous example
- must also have successfully completed Checkpoint 1

```
' The following format string variable displays two strings where the format
' for each is described by two comma-separated numbers enclosed between {x,y}
' where x describes which string is being formatted (0 is first string, 1 is
' second string, etc.) and y represents the number of characters allocated to
' the string (in the following example, we're using 10 characters). Note that
' a negative value indicates align left
Dim fmtStr As String = "{0,-10}{1,10}" & vbNewLine

' In the first example, we're displaying string "CSCI 130" in 10 characters
' left-justified and string "1:00pm" also in 10 characters but
' right-justified
outResults.AppendText(String.Format(fmtStr, "CSCI 130", "1:00pm"))
outResults.AppendText(String.Format(fmtStr, "CSCI 130L", "11:20am"))
```

The above code will produce the following output:

```
CSCI 130      1:00pm
CSCI 130L    11:20am
```

6. Use **File Explorer** to navigate to the folder containing the final version of your program `VideoRentalStore_Part5` in your lab folder. Copy and paste this program in the same folder, and then right-click on the copy to rename it as `VideoRentalStore_Part6`. Launch VS Express and use it to open the program `VideoRentalStore_Part6`.

   Enhance your program to allow users to purchase the following additional items during checkout: *candy* @ $1.50, *popcorn* @ $5.00 a bag, and *soda* @ $2.25.

Your program should behave identically to the solution which can be run by double-clicking the file `VideoRentalStore_Part6.exe` found in the `Executables` folder inside of your lab folder.

When done, save and close program `VideoRentalStore_Part6`.

Checkpoint 6 (85/100): A successful `VideoRentalStore_Part6` program:

- allows user to input number of candy, popcorn and soda units purchased and produces an itemized receipt formatted EXACTLY as shown in the example in the previous part
- must also have successfully completed Checkpoint 1

7. Use **File Explorer** to navigate to the folder containing the final version of your program `VideoRentalStore_Part6` in your lab folder. Copy and paste this program in the same folder, and then right-click on the copy to rename it as `VideoRentalStore_Part7`. Launch VS Express and use it to open the program `VideoRentalStore_Part7`.

   Enhance your program again so that the number of *sodas* purchased is not input via a textbox, but instead is controlled by the number of times that a button is pressed. For example, if the soda button is pressed six times, that would indicate that six sodas were purchased. Furthermore, each time that the soda button is pressed, a message should be displayed in the rich textbox designated for output indicating how many sodas have currently been sold to this customer. For example, a sale of three sodas, along with five regular videos and three premium videos rented for five days would result in the following output:

```
1 x Soda @ $2.25
2 x Soda @ $2.25
3 x Soda @ $2.25
===================================

Customer name: Mary Smith
Rental period: 5 days

Regular videos (5 @ $2.50)   $62.50
Premium videos (3 @ $5.25)   $78.75
Candy bars (0 @ $1.50)        $0.00
Popcorn (0 @ $5.00)           $0.00
Soda (3 @ $2.25)              $6.75
-----------------------------------
Subtotal                    $148.00
Tax                          $10.36
Total                       $158.36
```

   Keep in mind that variables declared within a subroutine are "created" and "destroyed" each time the button associated with the subroutine is pressed. Thus, any values that are assigned to the variable will be lost once the button's action has finished. Since the number of sodas variable needs to maintain its value between multiple button presses, that variable needs to be declared as a form-level (i.e. global) variable. If you use the variable name, `numberOfSodas`, it should be declared near the top of the form just below the class statement and above any subroutines. Recall that form-level variables must be declared here to be persist their values between button presses, as well as be available to all of the subroutines on the form. However, you should only declare variables as form-level if they need to be.

Your program should behave identically to the solution which can be run by double-clicking the file `VideoRentalStore_Part7.exe` found in the `Executables` folder inside of your lab folder.

When done, save and close program `VideoRentalStore_Part7`.

Checkpoint 7 (90/100): A successful `VideoRentalStore_Part7` program:

- allows user to press a soda button to indicate the sale of a soda item, instead of inputting the count of sodas via a textbox
- produces an itemized receipt formatted EXACTLY as shown in the previous example
- must also have successfully completed Checkpoint 1

8. Use **File Explorer** to navigate to the folder containing the final version of your program VideoRentalStore_Part7 in your lab folder. Copy and paste this program in the same folder, and then right-click on the copy to rename it as VideoRentalStore_Part8. Launch VS Express and use it to open the program VideoRentalStore_Part8.

   Complete your program by changing it so that the count of each item is controlled by pressing a button instead of inputting a number into a textbox. Likewise, each time an item button is pressed, a message indicating the current number of that item sold should be output to the rich textbox, as in the last part. Effectively, you will create a digital cash register where the buttons serve to "ring up" a customer. Your program should include buttons that correspond to each of the items for sale.

Your program should behave identically to the solution which can be run by double-clicking the file VideoRentalStore_Part8.exe found in the Executables folder inside of your lab folder.

When done, save and close program VideoRentalStore_Part8.

Checkpoint 8 (100/100): A successful VideoRentalStore_Part8 program:

- allows user to "ring up" a customer by pressing buttons indicating the the sale of an item
- produces an itemized receipt formatted EXACTLY as shown in the example in the previous part
- must also have successfully completed Checkpoint 1

Submission Instructions

Your M:\CS130\Labs\Lab03_YourLastName_YourFirstName folder should contain your solutions to this and Tuesday's lab.

To submit your work, copy this folder and paste it to N:/Handins/CS130/Lab03 PRIOR to the end of lab today. Late submissions won't be accepted and will receive a grade of 0 instead.