**Overview (1):**
- Before We Begin
  - Some administrative details
  - Some questions to consider
- Manipulation of Strings
  - Introduction
  - String related functions
  - Working with strings
  - Example -> Exercise 5-4

**Administrative Details (1):**
- Lab Exercises
  - You should be working on Ex 5-9 this week
    - Due March 6
  - We will be covering Ex. 5-4 and 5-5 in this weeks lecture
    - I recommend you work on these and the other exercises in Ch. 5 on your own

**Before We Begin**

**Some Questions to Consider (1):**
- What is a ListBox control?
- What is an item?
- How can we add items to a ListBox?
- How do we remove the information on a ListBox?
- Describe the Date type
- How can we manipulate Date types?

**Manipulation of Strings**
Introduction to Computer Use II

Review of Strings (1):

- Recall
  - A String is a sequence of characters enclosed between quotes
  - Characters are not restricted to being alpha-numeric (e.g., a,b,c... and 1,2,3...) → can be anything
  - Strings are of course widely used in many situations
    - Convey information (e.g., Label objects etc.)
    - User input is typically a string (e.g., TextBox)

Introduction (1):

- Often Need to Manipulate Strings
  - Given the widespread use of strings, there are many situations where we have to manipulate strings in some form or another
    - Convert lower-case to upper-case
    - Remove characters from a string
    - Add characters to a string
    - Concatenate two strings
  - Many string-related functions are available in Visual Basic (and many other programming languages) to perform a wide variety of operations on strings

String Related Functions (1):

- Some VB String-Related Functions
  - Visual Basic contains many built-in functions to perform a wide variety of operations on strings
  - See your textbook (page 5-17 and 5-18) for a listing of several of these functions
  - Let's take a look at a few of the more popular functions → remember, the best way to familiarize yourself with these functions is to practice using them!

String Related Functions (2):

- Some VB String-Related Functions (cont.)
  - InStr(Integer – optional, String1, String2)
    - Returns a Long specifying the position of the first occurrence of "String2" in "String1" from the beginning of "String1" (or from Start if optional argument is specified)
  - LCase(String1)
    - Returns "String1" converted to lower-case
  - UCase(String1)
    - Returns "String1" converted to upper-case

String Related Functions (3):

- Some VB String-Related Functions
  - Left(String1, Integer)
    - Returns a string containing the specified number of characters from the left of "String1"
  - Right(String1, Integer)
    - Returns a string containing the specified number of characters from the right of "String1"
  - Len(String1)
    - Returns a Long that specifies the number of characters the string contains

String Related Functions (4):

- Some VB String-Related Functions
  - StrReverse(String1)
    - Returns a string composed of the characters of "String1" but in reverse order
  - StrComp(String1, String2)
    - Returns an integer indicating the comparison of "String1" and "String2"
      - "String1" less than "String2" → -1
      - "String1" equal to "String2" → 0
      - "String1" greater than "String2" → 1

CSE 1530 Winter 2006
Bill Kapralos
**String Related Functions (5):**

- **Some VB String-Related Functions**
  - `Rtrim(String1)`
    - Returns a string with blanks removed from the right of "String1"
  - `Space(Long)`
    - Returns a string composed of just blanks as many specified by the Long argument
  - `Str(Long)`
    - Returns a string representation of the Long argument number

**Working With Strings (1):**

- **Some Notes**
  - Recall that a string is a sequence of characters
  - Beginning from the left of the string, we have the first character, second character etc...
  - You can think of each character in a unique position within the string → remember "index"?
  - Each character of the string has an index just as in control arrays but let's start at 1 not 0
  - First character → index 1
  - Second character → index 2
  - Third character → index 3 ...

**Working With Strings (1):**

- **Some Notes (cont.)**
  - Given we have this sequence of N characters, when we want to manipulate these characters in some manner, we can iterate through each character of the string using a loop!
  - Many times we use a loop to go through each of the characters in a string, examine them and potentially alter them
  - Since we know the length of the string (e.g., number of characters) a counted loop seems like the right choice!

**Example: Exercise 5-4 (1):**

- **Lets Practice Working With Strings**
  - Develop a program that requests the user to input a string and then perform some operation on the string depending on which option is selected by the user.

**Example: Exercise 5-4 (2):**

- **Lets Practice Working With Strings (cont.)**
  - Lets Look at the first option ("List all the Characters")
    - List each of the characters within the string in a ListBox, one character per line.

**Example: Exercise 5-4 (3):**

- **Lets Practice Working With Strings (cont.)**
  - Lets Look at the first option ("List all the Characters")
  - Steps to be performed
    1. Obtain the input string
    2. Obtain the length of the input string
    3. Set-up counted loop
      - Obtain character in the string → use the "Mid" function
      - Display character in ListBox

CSE 1530 Winter 2006
Bill Kapralos
Example: Exercise 5-4 (4):

- **Lets Practice Working With Strings (cont.)**
  - Let's look at the Mid function
  - **Mid(String, Start(Long), Length(Long - optional))**
  - Returns all (or "Length" if it is specified) characters from a string starting at position "Start"
  - Example → Assume following string: "Goodbye" and we want to obtain the fourth character "d"
  - `char = Mid("Goodbye", 4, 1)`

Example: Exercise 5-4 (3):

- **Lets Practice Working With Strings (cont.)**
  - Here is the Code to Perform the first operation
    ```vbnet
    inputStr = Text1.Text
    If (Option1(0).Value = True) Then
        ' Make the appropriate output control objects visible
        Label3.Visible = True
        Label3.Enabled = True
        List1.Visible = True
        List1.Enabled = True
        ' Let's now display the characters of the string
        For loopIndex = 1 To Len(inputStr)
            Dim char As String
            char = Mid(inputStr, loopIndex, 1)
            List1.AddItem (char)
        Next
    End If
    ```

- **Lets Practice Working With Strings (cont.)**
  - How should we proceed here?

Example: Exercise 5-4 (4):

- **Lets Practice Working With Strings (cont.)**
  - You should experiment with and at least complete some of the remaining options available
    - A good way to practice working with strings!
    - Work on them on your own and we can discuss the solutions to the rest of the options during the next lecture
    - Look at pages 5-24 and 5-25 in your textbook for tips on how to experiment (test) your program

As An Aside (1):

- **A Closer Look At Comparisons**
  - Be careful when comparing values and string!
  - May not always be what you think
  - Consider the following → we wish to compare two numbers (values): 50 and 100
  - Try the following on your own and try to understand the result → we will look at it next lecture...
    - `(50 < 100)` → True or False?
    - `("50" < "100")` → True or False?