CSE 1530

Introduction to Computer Use II: Programming

Winter 2006 (Section M)

Topic D: Control Structures - Iteration

Wednesday, March 1 2006

Bill Kapralos

CSE 1530, Winter 2006, Bill Kapralos

Overview (1):

- **Before We Begin**
  - Some administrative details
  - Some questions to consider
- **Manipulation of Strings (Cont.)**
  - Example Exercise 5-4
  - Example Exercise 5-5

Before We Begin

Administrative Details (1):

- **Lab Exercise**
  - You should be working on Ex 5-9 this week
    - Due March 6
  - Lab Ex. 4-8 has been graded and will be distributed after the lecture
  - I will be in the Glade Lab today, after the lecture from 2:30-3:30pm
- **Test 2 Reminder**
  - Wednesday, March 15 2006
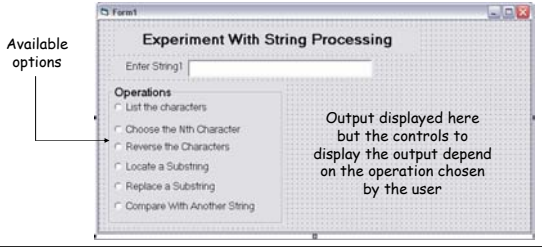 - Don't leave your studying until the last minute!

Some Questions to Consider (1):

- What is a string ?
- How can we determine the length of a string ?
- Describe the similarities/differencesw between a control array and a string
- How can we convert all characters of a string to lower/upper case ?
- How can we "reverse" a string ?
- Describe the "Mid" function

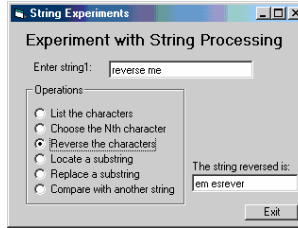# Manipulation of Strings (cont. from last lecture)

Example: Exercise 5-4 (1):

- ▣ **Review From Last Lecture**
  - ▫ Develop a program that requests the user to input a string and then perform some operation with the string depending on which option is selected

Available options →

Experiment With String Processing

Enter String1:

Operations
- ○ List the characters
- ○ Choose the Nth Character
- ○ Reverse the Characters
- ○ Locate a Substring
- ○ Replace a Substring
- ○ Compare With Another String

Output displayed here but the controls to display the output depend on the operation chosen by the user

Example: Exercise 5-4 (2):

- ▣ **Lets Practice Working With Strings (cont.)**
  - ▫ Lets Look at the third option ("Reverse the characters")
    - ▪ Simply call the StrReverse function

Experiment with String Processing

Enter string1: reverse me

Operations
- ○ List the characters
- ○ Choose the Nth character
- ● Reverse the characters
- ○ Locate a substring
- ○ Replace a substring
- ○ Compare with another string

The string reversed is:
em esrever

Exit

But what about not calling the StrReverse function? Can you think of a way to perform this task without actually calling the function?
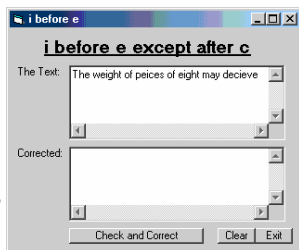
Example: Exercise 5-4 (3):

- ▣ **Lets Practice Working With Strings (cont.)**
  - ▫ Lets Look at the fourth, fifth and sixth options ("Locate a sub-string")
    - ▪ Will have to prompt the user for a second string
    - ▪ Once the appropriate input has been entered, you can call the appropriate functions to perform the required task

Example: Exercise 5-5 (1):

- ▣ **Spell Checking – The I Before E Rule**
  - ▫ Develop a program to correct the spelling of words containing "ie" or "ei"
    - ▪ Rule → "i before e except after c"
    - ▪ Keep in mind the purpose of this exercise is to practice with loops and strings and manipulating strings → the output of this program may not necessarily be correct gramatically!

Example: Exercise 5-5 (2):

- ▣ **Spell Checking – The I Before E Rule (cont.)**
  - ▫ Take a look at the user interface below
    - ▪ Text is entered in a Textbox not ListBox!
  - ▫ User enters text in the upper TextBox & when the "Check and Correct" button is pressed, the corrected version appears in the lower TextBox

i before e except after c

The Text: The weight of peices of eight may decieve

Corrected:

Check and Correct   Clear   Exit

Example: Exercise 5-5 (3):

- ▣ **Adding a Scrollbar to a TextBox**
  - ▫ In this exercise the user will enter the text in the upper TextBox and the corrected text will be displayed in the lower TextBox
    - ▪ But what if the text cannot fit in the TextBox ? → we can add a scroll bar to the TextBox
  - ▫ The TextBox object has a property allowing it to have and display scrollbars!
    - ▪ Set the MultiLine property to True
    - ▪ Set the Scrollbar property to 3- Both

Example: Exercise 5-5 (4):

- **Adding a Scrollbar to a TextBox (cont.)**

MultiLine property → must be set to True

TextBox control

Scroll property → must be set to Both
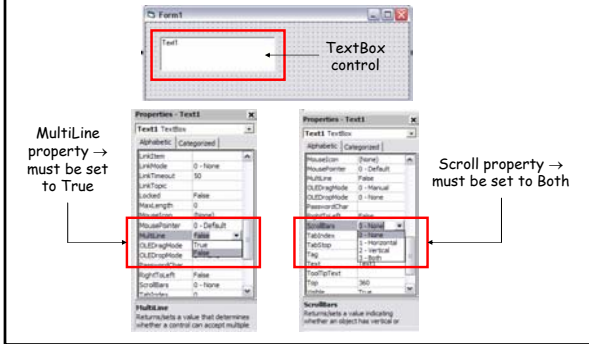
Example: Exercise 5-5 (5):

- **Overview of the Solution**
  - Declare a String variable and assign it the value from the input textbox
  - Set up a loop that checks for the occurrence of "ei" in the input string
    - If the occurrence of "ei" is not preceded by "c", replace it by "ie"
    - Continue until all occurrences of "ei" have been examined and potentially altered
    - Assign the corrected string to the output TextBox

Example: Exercise 5-5 (6):

- **Lets Examine the Required Loop**
  - We do not know how many times the letters "ei" occur within our input string,
    - Therefore, we will need a conditional loop
    - Keep in mind that there may not even be a single occurrence of "ei" in the string therefore, we need a Do/While loop → if "ei" does not appear, then we do not want to iterate loop at all!
  - What condition will we test for ?
    - Before we decide this, lets look at how we will go about solving the problem in greater detail

Example: Exercise 5-5 (7):

- **Lets Examine the Required Loop (cont.)**
  - What functions do we have available that will help us in solving our problem ?
  - InStr(Start(Integer, optional), String1, String2)
    - This function returns a Long specifying the position of the first occurrence of "String2" in "String1", starting at "Start" if specified, otherwise at the beginning of "String1" (e.g., at character 1)
    - If no occurrence of "String2" in "String1" then 0 is returned

Example: Exercise 5-5 (8):

- **Lets Examine the Required Loop (cont.)**
  - How about the following condition for our conditional loop → InStr(txtInput, "ei") > 0
    - Assuming "ei" does not appear the input string (denoted by "txtInput"), then the loop will not execute at all
      If it does occur then it will execute but as written above, it will never end since it will always find the first occurrence of "ei" within txtInput
    - Need to find the first occurrence, then the second etc.

Example: Exercise 5-5 (9):

- **Lets Examine the Required Loop (cont.)**
  - But remember that the InStr function allows us to specify an argument indicating where in the input string the search should begin!
  - Consider the following code:

```
position = 1
Do While (InStr(position, txtInput, "ei") > 0)
    position = InStr(position, txtInput, "ei")
    char = Mid(txtInput, position – 1, 1)
    result = StrComp(char, "c") + StrComp(char, "C")
    If (result = 0) Then
        ' Perform the necessary actions here to replace "ei" with "ie"
        …
    End If
Loop
```

Example: Exercise 5-5 (10):

- **Lets Examine the Required Loop (cont.)**
  - Consider the following input: "Conceive a piece plan"
  - On the first iteration, position has the value 1
  - First statement within the loop assigns five to it
  - Statement following "If" makes position = 6
  - During second iteration, the InStr functions looks for next "ei" occurring after character 6.
    - Returns 13 and the loop body is executed again
    - Position is assigned 13, then after the If block it is assigned 14 and repeats again → loop condition will now be False since InStr returns 0

Example: Exercise 5-5 (11):

- **Replacing One Occurrence of the Substring**
  - We want to now replace (where appropriate) the occurrence of "ei" with "ie" → once again, lets consider the functions we have available to us
  - Replace(String1, String2, String3, Start(Long - optional), Count (optional))
    - Returns a string with "String2" replaced by "String3" where ever it is found in "String1", beginning at the position "Start" or replaces it "Count" times from position Start
    - If Start is not specified then 0 is assumed

Example: Exercise 5-5 (12):

- **Replacing An Occurrence of the Substring**
  - Must be careful when we use the Replace function since, if used as shown below, all occurrences of "ei" will be replaced by "ie"
    - txtInput = Replace(txtInput, "ei", "ie")
    - May not necessarily be what we want! Therefore, replacement needs to occur one at a time
  - What about the following
    - Replace(txtInput, "ei", "ie", position, 1) ?
    - Still not sufficient! → Will truncate characters in return string that are before "Start"

Example: Exercise 5-5 (13):

- **Replacing An Occurrence of the Substring**
  - So if characters occurring before "position" are truncated, how can we perform the intended task ?
    - Use the Mid function → Obtain characters before "position" using the Mid function then catenate them to result of calling Replace
  - See course website for complete solution to this problem

Example: Exercise 5-5 (14):

- **Putting it All Together**
  - You should now be able to, on your own, complete Ex. 5-5 and in fact, it's a good idea to actually do so
    - Good preparation for Ex. 5-9 that must be submitted
  - Of course, once you complete the program, test it with various input strings
  - I will make my solution available online