Inft2012 Application Programming – Notes for week 8

Lab exercises

You might choose to spend this week's lab time working on the assignment. Even so, here are some exercises relevant to this week's lecture.

1. Write a program that creates, say, half a million random five-letter strings. Each string will be five randomly selected lower-case letters. It might be a word, such as simon, index, array, place, kazoo, or orris, but it's more likely not to be. Each time you create a string, add it to an array of string and to a list of string.

   Add a textbox to the form, and two buttons: one to find and delete from the array the string in the text box, and one to find and delete from the list the string in the text box. To delete a string from the array, you will need to move every subsequent element down one place, then make the last element the empty string. Do you appreciate the simplicity of deletion from a list?

   To confirm that the deletions are working, it might help if you can see one or more of the strings in the array and/or the list. One way of doing this would be to add a 'Show at index' button: you enter an index in the text box, click the button, and a message box displays the entry at that index.

2. Write a program that opens a text file and copies its contents into a multiline text box.

   Add a 'Code' button that removes all non-letter characters, turns all letters to upper case, and displays them in another text box in blocks of five separated by spaces (a common representation in the days of radio transmission of encrypted text). For example, the beginning of this paragraph would be represented as ADDAC ODEBU TTONT HATRE MOVES. How easy is it to read text in this form?

3. Create a text file with people's names, one per line. Put as many or as few names as you wish.

   Write a program that will read the names from the file and store them as a list of strings. You will not know the number of names in advance, so you should read the file line by line, checking for end of file. The program should also be able to display the names in the list; to sort the names alphabetically; to reverse the set of names; to add names; to delete names.

   The program should also be able to save the possible amended list of names as a text file. But it should also be able to save the names as a binary file, and to read the binary file, putting the names in a list.

   If you're not convinced that a binary file is different form a text file, (1) examine the binary file in a text editor; (2) close Visual Studio, open it again, and start from the binary file, before any text has been read.