GDAPS2 – Practice Exercise

Linked Lists, Part 1

# Objective

Practice implementing parts of a Linked List. You will be building upon this exercise in the next class, and then using it as a starting point for the first homework assignment.

*Note: C# comes with a built-in LinkedList class – Do* ***NOT*** *use it!*

*You’re creating your own for this exercise.*

# Details

**CustomLinkedNode Class**

A linked list is a collection of nodes. Each node in a linked list is more than a single piece of data; it also holds a link to the next node. This means you’ll need to create a generic CustomLinkedNode class which holds those two things: the raw data it’s storing, as well as a link (reference) to the next node in the list (if there is one).

Make sure you have appropriate get/set properties for both the data and the next node.

**CustomLinkedList Class**

Now that you have the nodes, you can begin implementing the generic linked list itself. It should track the total amount of data with a count field (and get property for the count).

As far as the nodes, at a minimum you only need a reference to the head node, as all other nodes can be found by traversing the list. It may be beneficial to also have a reference to the tail node, as that can simplify adding new data to the list.

Add 2 methods to your CustomLinkedList class: Add and GetData.

**void Add(T data)**

The add method should accept data of the generic type and add it to the end of the list. Remember that all data is stored in a node, so first create a CustomLinkedNode and then “hook it up” to the end of the list. Don’t forget to adjust the count as well.

**T GetData(int index)**

This method should return the *data* (not the whole node!) at the specified index. If the index is invalid, throw an exception instead. Finding data somewhere in the middle of a linked list requires looping through the list, starting from the head node. (This could be an indexer property instead.)

# Main Method

Create a CustomLinkedList object and ask the user to add at least 5 pieces of data to it. (The data type and content of your linked list is up to you.) Then print the count of the list, to ensure that the Add method is working.

Loop through the list and print each item to ensure your GetData method is working.

Note: You won’t be able to use a foreach loop with your Linked List, since

your class doesn’t implement the IEnumerable interface.

# Sample Run

What are you adding to the list? arrow

What are you adding to the list? boots

What are you adding to the list? cloak

What are you adding to the list? diary

What are you adding to the list? essence of fire

Here is your list:

arrow

boots

cloak

diary

essence of fire

# Submission

All of your work must be commented and follow this course’s coding standards. **Read through the Coding Standards document (located in MyCourses) to check over your code before you complete your program. Make sure you follow the coding standards for all code you create.**

1) Submit: Submit your program to the appropriate Assignments dropbox in MyCourses.

2) Check-off: Show your working program to the instructor or TA. If you do not finish before class ends, complete the exercise for homework and show one of us in-class on the next class period. If your program works as expected, you will be “checked off” to earn credit for the exercise.