

CS 112 Lab Practicum

October 31 2025

Linked Lists and the StringNode Class

In lecture, we've been considering linked lists of characters, where each node in the linked list is an instance of the following class:

```
public class StringNode {
    private char ch;
    private StringNode next;

    //Constructor
    public StringNode(char c, StringNode n) {
        this.ch = c;
        this.next = n;
    }
    ...
}
```

Our goal for this practicum is to take a String as an input, and then return the head of a linked list string where the characters are in alphabetical order.

If the input String contains a capital letter, it must be converted to a lowercase letter when creating the linked list string. If there are non-letter characters, those should be removed from the linked list String.

Examples:

- input string: "backtrack"; returned linked list string: "aabckkrt".
- input string: "Hello World!"; returned linked list string: "dehllloorw".
- input string: "?!"; returned linked list string: null.
- input string: null; returned linked list string: null.

We will do this by creating two methods in the StringNode class

1. A method called **insertSorted()**, which takes in a character and the head of a linked list string. Under the assumption that the linked list is already in sorted (ie alphabetical) order, this method will insert the provided character into the linked list while maintaining the alphabetical order. This method will return the head node of the updated linked list string.
2. A method called **convertAlphabetical()** which takes in a String and performs the conversion outlined in the above examples. This method will return the head node of the linked list string. Make use of the **insertSorted()** method!

Note 1: You may use **isLetter(char c)** method from the **Character** class. This method is static and returns a boolean value.

Note 2: You may use the **toLowerCase()** method from the **String** class. This method is non-static and returns a String.

```

public class StringNode {
    private char ch;
    private StringNode next;

    //Constructor
    public StringNode(char c, StringNode n) {
        this.ch = c;
        this.next = n;
    }

    ...
    //other methods
    ...

    //Write your methods here (1 & 2)

```

}