Object Oriented Programming
Lab 04 Linked List

February 9, 2023

1 Lab Goal

The purpose of this lab is to gain a basic understanding of Lists that you will use while working with Java. This is a Data Structures topic but its importance cannot be trivialized when it comes to Object Oriented Programming. You need to have a good understanding of objects and references when using Objects with lists and helping you make your own lists helps you in understanding just that.

You can access Java Documentation at this link.

1.1 Making a Linked List

The code below helps you in understanding how a basic linked list is created. We will be using this as the base for all the lab problems to be done. You can check out its code below:

```
1
2  class Node {
3      int data;
4      Node next;
5
6      public Node(int data) {
7          this.data = data;
8          this.next = null;
9      }
10 }
11
12 class LinkedList {
13     Node head;
14
15     public void append(int data) {
16         if (head == null) {
17             head = new Node(data);
18             return;
19         }
20         Node current = head;
21         while (current.next != null) {
22             current = current.next;
23         }
24         current.next = new Node(data);
25     }
26
27     public void remove(int data) {
28         if (head == null) {
29             return;
30         }
31         if (head.data == data) {
32             head = head.next;
33             System.out.println("removed");
34             return;
```

```
35          }
36          Node current = head;
37          while (current.next != null) {
38              if (current.next.data == data) {
39                  current.next = current.next.next;
40                  return;
41              }
42              current = current.next;
43          }
44      }
45  }
46
47  public class LinkedListManager
48  {
49    public static void main(String[] args)
50    {
51      LinkedList l = new LinkedList();
52      l.append(0);
53      l.remove(0);
54    }
55  }
```

2 Problems

We will start with a few basic problems and then later move on to a bit complex ones.

2.1 Problem 1 - Creating separate classes

The first problem is to use the code given above and create separate class files for all the classes and make the code work.

2.2 Problem 2 - Expanding the functionality

In this problem you will add the following functions/functionality:

- Add a function by which you can add a number to a specific index. If the index does not exist, the number should be added to the tail of the list. If the index is in the middle of the list, then all the numbers will shift a place down.
- Add a function that removes a number from a specific index value. Keep in mind that the number may be present at the start, end or middle of the list.
- Add a function that removes all the occurrences of a certain number from the list
- Add a function that displays the current state of the list with all the number entries

2.3 Problem 3 - Storing Strings

Write a function that accepts a String as the input value and then it extracts the ASCII value of every character of the passed string and stores it in the list that you created.

2.4 Problem 4 - Retrieving Strings

Write a function that returns a string by retrieving all the ASCII values of the stored characters and then combines the characters together to form a String