Lab Report 01

*Lian Liao*

**Problem**  
Write a class called **DoubleDoubleLL** which is a double linked list of type double.  This link list is similar to the single linked list that was shown in class except that each node in addition to having **data**and **nextLink**(which was originally called **link**) now has **prevLink.**

**Solution**

First of all, I tried to understand how driver test. Then followed the requirements create internal class and its instance variable with constructors. They represent the content in current. In outside class, declared head and current as node of list. Finally, made all the method it need. The three important method I think is add, addAfterCurrent, and removeCurrent.

**Implementation Problems Encountered**

I got one error in the beginning, I wrote current.nextLink.prevLink instead of current.prevLink.nextLink. It took me some time to find it.

**Lab Report Questions**

1.If the reference to an object is lost and thus the object is now unreachable, what does the Java Virtual Machine (JVM) do with said object?

Once an object is no longer referenced and therefore is not reachable by the application code, the garbage collector removes it and reclaims the unused memory.

2.What are some advantages and disadvantages of using a doubly linked list versus a singly linked list?

Disadvantages: a. It uses extra memory when compared to the array and singly linked list. B. Since elements in memory are stored randomly, therefore the elements are accessed sequentially no direct access is allowed.

Advantages: a. It can allocate or reallocate memory easily during its execution. b. A singly linked list deletion requires a pointer to the node and previous node to be deleted but in the doubly linked list, it only required the pointer which is to be deleted.