CS62: Spring 2025 | Lecture #8 (Linked Lists) worksheet | Jingyi Li

1. Suppose x is a reference to a Node and that node is not the last one on the singly linked list. What is the effect of the following code fragment?

   ```
   x.next = x.next.next;
   ```

2. Suppose x and t are references to different Nodes in a singly linked list. What is the effect of the following code fragment?

   ```
   t.next = x.next;
   x.next = t;
   ```

3. Suppose x and t are references to different Nodes in a doubly linked list. What is the effect of the following code fragment?

   ```
   t.prev = x;
   t.next = x.next;
   x.next.prev = t;
   x.next = t;
   ```

4. What if instead the code was in a different order, i.e.:

   ```
   x.next = t;
   x.next.prev = t;
   t.next = x.next;
   t.prev = x;
   ```

```
/** 5. Write the get method for doubly linked lists:
     * Returns element at the specified index.
     *
     * @param index the index of the element to be returned
     * @return the element at specified index
     * @pre: 0<=index<size
     */
    public E get(int index) {
       // check if index is valid



       // if index is 0, return element at head




        // else if index is size-1, return element at tail




         // set a temporary pointer to the head




         // search for index-th element or end of list



         // return the element stored in the node that the temporary pointer points to


    }

    /** 6. Write the addLast method for doubly linked lists:

     * Inserts the specified element at the tail of the doubly linked list.
     *
     * @param element the element to be inserted
     */
    public void addLast(E element) {
        // Create a pointer to tail


        // Make a new node and assign it to tail. Fix pointers.






        // if first node to be added, adjust head to it.



        // else fix next pointer to tail



        // increase number of nodes


    }
```