Wyo Data Structures Name –
Ch. 19 Worksheet #11

Implement the following methods as they would need to be implemented in the LinkedList class that we have stored in our Handout folder.

Implement the two methods in bold. Print your answers on a separate piece of paper.

```
import java.util.NoSuchElementException;

public class LinkedList
{
	public LinkedList(){    first = null;  }

	public Object getFirst() {
		if (first == null)	throw new NoSuchElementException();
		return first.getValue();
	}

	public void addFirst(Object obj) {
		ListNode newNode = new ListNode(obj, first);
		first = newNode;
	}

	public Object removeFirst() {
		if (first == null)	throw new NoSuchElementException();
		Object obj = first.getValue();
		first = first.getNext();
		return obj;
	}

	public Object getLast() {
		if (first == null)	throw new NoSuchElementException();
		ListNode temp = first;
		while (temp.getNext() != null)
			temp = temp.getNext();
		return temp.getValue();
	}

	public void addLast(Object obj) {
		ListNode newNode = new ListNode(obj, null);
		ListNode temp = first;
		while (temp.getNext() != null)
			temp = temp.getNext();
		temp.setNext(newNode);
	}

	public Object removeLast() {
		if (first == null)	throw new NoSuchElementException();
		if (first.getNext() == null)
		{
			Object obj = first.getValue();
			first = null;
			return obj;
		}
		ListNode temp = first;
		ListNode follower = temp;
		while (temp.getNext() != null)
		{
```

```
                follower = temp;
                temp = temp.getNext();
            }
            follower.setNext(null);
            return temp.getValue();
        }

        public boolean find(Object key) {
                // returns true if key is found in the linked list, false otherwise
        }

        public ListNode findMiddle() {
                // if there is an odd number of nodes in the linked list,
                // a reference to the middle ListNode is returned. If there
                // is an even number of nodes, then a reference to the ListNode
                // that comes right before the middle element is returned (for
                // example if there are 5 nodes, then a reference the 3rd node
                // is returned. If there are 4 nodes, then a reference to the 2nd
                // node is returned.
        }

        private ListNode first;
}

--------------------------------------------------------------------------------

public class ListNode
{
        public ListNode(Object initValue, ListNode initNext) {
            value = initValue;
            next = initNext;
        }

        public Object getValue() { return value; }

        public ListNode getNext() { return next; }

        public void setValue(Object theNewValue) { value = theNewValue; }

        public void setNext(ListNode theNewNext) { next = theNewNext; }

        private Object value;
        private ListNode next;
}
```