Lesson-2 Class Notes and Review points

Interface & Abstract Classes are useful to create ADT in implementation.

If I want to prepare a Shopping List what kind of operations you can do

ShoppingList ADT – Includes operations and data type

```
 double CalculateTotalPrice();
        addItem(String item);
        removeItem(pos);
        clear();
        size();
        elementAtIndex(index) throw Exception
```

Linked List

1. Single Linked List
2. Doubly Linked List
3. Circular Linked List(Single & Double LL)

Position based List ADT operations – Can use with Array and Linked List

Here position may be an index or node reference.

- Generic methods:
  - size() return integer , isEmpty() return boolean

- Query methods – return boolean
  - isFirst(p), isLast(p)

- Accessor methods: To retrieve/getter
  - first(), last() → return the first and last element
  - before(p), after(p) → return the element before or after p element you passed.

- Update methods:-
  - replaceElement(p, e) → Modify the elements like setter
  - swapElements(p, q) → swap p and q elements, shift the values
  - insertBefore(p, e) → insert e before p position
  - insertAfter(p, e)→ Insert e after p position
  - insertFirst(e) → Insert e in the first position
  - insertLast(e) → Insert e in the last position
  - remove(p) → remove the element p

Example:

You have the Doubly List Implementation with position-based ADT operations.

Problem: Write an algorithm to get the sum of elements in the Doubly Linked list

```
Algorithm sum(L)
   Input: Doubly Linked List
   Output: Sum of all the elements in the given list
       if L.isEmpty() then return 0
       p := L.first()  // Get the first node
       sum := p.element() // get the value present in the first node
       while ! L.isLast(p) do // loop through until reach the last node
               p := L.after(p) // Move the next node after p
               sum := sum + p.element() // get the value present in the p node
       return sum // after the loop, return the computed sum
```

What is the time complexity of Doubly linked list sum algorithm?

Need to process each element in the list takes O(n) time to sum.

Use Iterators to print the List elements

```
objectIterator  It = L.elements();
While(it.hasNext())

   P = it.nextObject();
  Log.console(p.element())
```

// Summation using Iterator

```
Sum = 0
objectIterator  It = L.elements();
While(it.hasNext()) // More elements
    P= it.nextObject(); // Move to Node pointer , you got the first node. If the loop continue it
                        //will automatically move the next node, until loop fails
    Sum = sum + p.element()
```

__**Review Questions for Lesson-2**__

1. Able to write a pseudo code for the sequence ADT methods given in the slide 12. Refer Lab 2 Java script code given with the assignments.
2. Able to analyze the performance of array based implementations of sequence data structure.
3. Write the Node structure of Single and Double LL in JavaScript.

4. Able to know how to work with Linked List ADT and its operations in pseudo code.
5. How to analyze the performance of the Linked List ADT.
6. Able to solve the given problem using Slide 22 methods. Refer Slide 22-24.