

## Lab-5

Date: 09/02/2018

Design a data structure in Java to represent and manage polynomials ( $f(x)$ ) of any order. A polynomial can be treated as a list of term objects, where each term object represents coefficient and the exponent. You can use generic Linked List from Java generic collection to store these term objects. The following operations should also be supported in your implementation?

- Evaluate a polynomial for a specified value of  $x$
- Differentiation of a polynomial
- Addition of two polynomials
- toString method to display the polynomial

**Additional Functionality: (No marks for this part can be solved on self interest) Provide the following additional operations on polynomials**

- Multiplying a polynomial with a constant
- Subtraction of two polynomials
- Multiplication of two polynomials

### Notes on using Java Iterator for LinkedList:

When you need to access elements of the linked list in the sequence and perform some operation on the elements, using the get() method of the linked list will be costly, as, the get() method traverses the list every time it is called.

Another alternative is to use Java Iterator. Iterator facilitates to traverse (access) elements of the list one by one. Iterator is an interface provided by Java with the following methods.

- hasNext() – returns true if more unvisited elements are there in the list or collection
- next() – returns the next unvisited element in the list or the collection

**Example:** The following code demonstrates the usage of Iterator for LinkedList

```
//Imports
import java.util.LinkedList;
import java.util.Iterator;

LinkedList<Integer> list = new LinkedList<Integer>();
//add elements in the list
list.addFirst(1);
list.addFirst(2);
list.addFirst(3);
Iterator<Integer> iter = list.listIterator(0);
while(iter.hasNext())
{
    Integer ele = iter.next();
    //code to process the element
    System.out.print(ele + "->");
}
System.out.println("X");
```