Week 4 – Lecture 1

Linked List Problems Solving

Note: To solve the three exercises below use the LinkedList implementation (posted to the classroom) which contains the below classes:

- NodeData
- Node
- LinkedList

Exercise 1

Add a method in the class LinkedList named **sumValues( )** that returns the sum of the values in all nodes of a linked list.

Create an application in which you:

1. Create a linked list and add to it the values 6, 4, 10, 9, 1.
2. Call the method **sumValues** to output the sum of the values in the linked list (the sum will be 30).

Exercise 2

Add a method in the application named **sumNumbers( )** that receives as a parameter from the main method a linked list and returns the sum of the values in all nodes.

Create an application in which you:

1. Create a linked list and add to it the values 6, 4, 10, 9, 1.
2. Call the method **sumNumbers** to output the sum of the values in the linked list (the sum will be 30).

Exercise 3

Add a method in the application named **divisibleByN(LinkedList list1, int n)** that takes as parameters from the main a linked list **list1** and and an integer **n**, and returns the number (counter) of nodes that hold a value that is divisible by n.

Create an application in which you:

1. Create a linked list and add to it the values 6, 4, 10, 9, 1.
2. Read an integer n from the user.
3. Call the method **divisibleByN** to output the number (counter) of nodes that hold a value that is divisible by n.

For example, for the above list if n is 3 then the method will return 2 since only 6 and 9 are divisible by 3.