University of Victoria
Computer Science

Lab - 3

CSC115: Fundamentals of Programming: II

Fall 2016

Goal

- **Sign up for the lab**
- How to implement an interface
- How to implement a singly linked list
- Submit one file: *SLinkedList.java* at the end of the lab. **Click the “Submit” button**

1. INode.java

- Note that INode.java is a singly linked node. It is the building block for a singly linked list

2. Interface

- An interface provides a way to specify methods and constants, but supplies no implementations. There are two reasons to use interface: one reason is for different performances. For example, the IntegerList interface in assignment 2 and in today's lab is implemented in three classes: array based (assignment 2), singly linked list (lab 3) and doubly linked list (assignment 2). The performance of the addFront() method differs among them. Why? The second reason to use an interface is that it enables you to specify some desired common behavior that may be useful over many different types of objects. For example, in the assignment 2, the IntegerList is implemented differently, but as a programmer, you can write a program using the interface. The same algorithm applies to all the implementations.

- Learn to write simple program to test your code. Start from the constructor.
- The lab instructor is going to draw a lot of diagrams on the whiteboard and explain how a singly linked list works, how an INode object is manipulated.

The "javadoc" style comments are added to INode.java. To learn more, go to:

http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html#examples

Use Connex to submit one file: *SLinkedList.java* at the end of the lab. Make sure you click the "Submit" button.