**Lab Assignment #1 – Using Fundamental Data Structures**

Due Date: Friday, Week 3

Purpose: The purpose of this Lab assignment is to:

1. Design and develop Applications that incorporate fundamental data structures such as:

* Singly Linked Lists
* Doubly Linked Lists
* Circularly Linked Lists

References: Read the course’s text chapter 3 and the lecture slides. This material provides the necessary information that you need to complete the exercises.

Be sure to read the following general instructions carefully:

- This assignment must be completed individually by all the students.

- You will have to **demonstrate your solution in a scheduled lab session** and upload the solution on **eCentennial** through the assignment link.

**Exercise 1**

In this exercise, you will use the *DoublyLinkedList* implementation of the textbook (week 2 lecture examples. Write a method for concatenating two doubly linked lists L and M, with header and trailer sentinel nodes, into a single list L′. Write a main method to test the new method. **Hint**: Connect the end of L into the beginning of M.

A screen shot of a computer program

AI-generated content may be incorrect. (3 marks)

**A screenshot of a computer

AI-generated content may be incorrect.**

**Exercise 2**

In this exercise, you will add a method *swapNodes* to *SinglyLinkedList* class from week 2 lecture examples. This method should swap two nodes *node1* and *node2* (and not just their contents) given references only to *node1* and *node2*. The new method should check if *node1* and *node2* are the same node, etc. Write the main method to test the *swapNodes* method. **Hint**: You may need to traverse the list.

(4 marks)

**Exercise 3**

In this exercise you will add a *clone()* method to *CircularlyLinkedList* class from week 3 lecture examples. Write a main method to test the new method. **Hint**: See the implementation of clone method in *SinglyLinkedList* class, Make sure to properly link the new chain of nodes.

(3 marks)

**Evaluation:**

|  |  |
| --- | --- |
| **Functionality:**  Correct implementation of requirements  Code explanation when asked | 85% |
| **Object-Oriented design**: correct design of classes and methods similarly to chapter 3 examples. | 15% |
| **Total** | 100% |

You must name your Eclipse project according to the following rule:

**YourFullname\_COMP254Labnumber\_Exercisenumber**.

Example: **JohnSmith\_ COMP254Lab1\_Ex1**

**Submission rules:**

Submit your modules as **zip files** that are named according to the following rule:

**YourFullname\_ COMP254Labnumber\_Exercisenumber.zip**

Example: **JohnSmith\_ COMP254Lab1\_Ex1.zip**

Use 7-zip to compress files (https://www.7-zip.org/download.html).