Submission Guidelines:

⦁ All the labs are already uploaded on E-Learning, download and attempt.

⦁ Each topic should be started on new page.

⦁ Topic should be given Heading 1 with Calibri Light, 16 font size, Bold

⦁ Each Task should be given Heading 2 with Garamond, 14 font size, Bold

⦁ Each Problem should be in Calibri Body, 12 font size, Bold

⦁ Each Solution (code) should be in Consolas, 10 font size

⦁ Each solution should be either you have already submitted on E-Learning or should be your code.

⦁ Submit two files (word and its PDF version file) with the specified format.

⦁ Each student has to do assignment individually.

⦁ Ensure the correctness of code (text format).

⦁ Use appropriate headings and labels for each task and topic.

⦁ Follow the order of topics and tasks as mentioned in the assignment.

⦁ Plagiarism is strictly prohibited. If caught, zero marks will be assigned for the entire assignment for all involved students.

⦁ No Late submission (via email/WhatsApp/other platform) is allowed.

⦁ This assignment is OPTIONAL, if you attempt it, you will get bonus marks in sessional.

Note:

⦁ Make sure to replace placeholders (like "Your Name," "Student ID," etc.) with your actual information.

⦁ Provide complete and working code for each task.

⦁ Include any additional explanations or comments if needed for better understanding.

# **Lab 01- Arrays/LinkedList**

Task01 (NLP)

Create a file named NLArray.java and design following functions:

- String [] wordTokenize (String fileName) à Read any text file and return list

of words from that file. (Ignore . , : and all these types operators)

- String[] extractEmail (String fileName) à Read any text file and return all

emails from and file

Note: Read about Natural Language Processing (NLP), Word Tokenizing, Stop Words,

Information Extraction/Retrieval for Knowledge.

Task02 (Image Cropping)

Design following methods in above same class NLArray.java.

- void extractBoundaries (int arr[][]) à This function should extract

boundaries and print from arr (Boundaries include 1st row, 1st col, last row, last col).

- void cropCenterPart (int arr[][]) à This function should extract center part

and print from arr, center part includes everything except Boundaries (Boundaries include

1st row, 1st col, last row, last col).

Task03 (Determining N consecutive same values)

Design following methods in above same class NLArray.java.

boolean NConRep (int arr[][]) à This function would return True if N consecutive values are

same otherwise false. Check examples:

2 1 3 5

22 22 22 22

12 41 88 53

57 8 74 4

N is 4 so matrix is 4x4 and in 4 consecutive values are same so it should return True.

Lab01 – Arrays/Linkedlist Data Structures Instructor: Saif Hassan

Page 3 of 4

Task04 (Linked Lists)

In this task you will write a program that implements a variant of a linked list. This variant has a dummy

node pointed to by the head link as shown in the following figure:

Linked list with a dummy first node:

This trick will allow your code to be a little simpler, not requiring a special case for add or remove

operations. Your constructor method will be:

public LinkedList(){

head = new Node(null);

size = 0;

}

You need to write a class called LinkedList that implements the following List interface:

// a list interface

public interface List {

public boolean isEmpty();

// returns true if the list is empty, false otherwise

public int size();

// returns the number of items in the list

public void add(Object item);

// adds an item to the list

// item is added at the end of the list

public void add(int index, Object item);

// adds an item to the list at the given index

// item is added at the given index; // the

indices start from 1.

public void remove(int index);

// removes the item from the list that has the given index

public void remove(Object item);

// removes an item from the list

// removes the first item in the list whose equal method matches

Lab01 – Arrays/Linkedlist Data Structures Instructor: Saif Hassan

Page 4 of 4

// that of the given item

public List duplicate();

// creates a duplicate of the list

// returns a copy of the linked list

public List duplicateReversed();

// creates a duplicate of the list with the nodes in reverse order

// returns a copy of the linked list with the nodes in reverse order

}

In addition to the interface, your LinkedList class needs to implement a toString() method that

prints the list in the format

[ size: the\_size\_of\_the\_list - item1, item2, .... ]

Specifications, notes, and hints

Your program needs to meet the following specifications:

• Submit the file LinkedList.java and additional files if applicable. Your Node class should be

an inner class within the LinkedList class. Make sure your class implements the interface

as specified, i.e. your class should begin with public class LinkedList implements

List.

• When commenting your code use Javadoc style comments at the beginning of each method.

• Put comments at the top of the file (Java File) with your name, S\_ID, S\_Name, date and course,

and a short (one or two line) description of what the program does. Make sure your code runs on

machine.