COMP 202: Data Structures and Algorithms
Lab work 2: Implementing Stack Data Structure

1 Purpose

To implement linear data structures.

In this exercise, you will implement stack data structure. You are free to implement them in your preferred programming language.

2 Background

Linear data structures organize data elements in a linear manner, i.e. each data element has only unique successor. Array, stack, queue, linked lists etc. are some examples of linear data structures.

2.1 Stack

A *stack* is a linear data structure where all insertions and deletions are restricted to one end, called the *top*. It is also known as a *Last-In-First-Out (LIFO)* list because the last element inserted into a stack is the first element removed.

Principal operations on a stack are *push* (adding an element into a stack) and *pop* (removing an element from a stack).

3 Tasks

1. Implement Stack data structure using an array as well as a linekd list. Following operations must be available:

   (a) push(element): Adds an element into the stack

   (b) pop(): Removes an element from the stack

   (c) isEmpty(): Checks if the stack is empty

   (d) isFull(): Checks if the stack is full

   (e) top(): Gives the element at the top

   Also, write a test program to check if the stack implementations work properly.

4 Lab work submission

Submit your work via Canvas **within 2 weeks**. Your submission must include the following:

1. Your code
2. A report containing
   (a) the output of your program, and
   (b) answers to the questions posed in the labsheet, if any.
3. Link to your Git repository must be in a comment. Your repository must be private. The user whom you must give access will be communicated during the lab session.