COMSATS UNIVERSITY ISLAMABAD (CUI)

DEPARTMENT OF COMPUTER SCIENCE
TERMINAL EXAMINATION FALL - 2023
BS (CT) 3A-A SEMESTER

Course: CSC211-Data Structures & Algorithms **Instructor:** *Zahid Anwar*
Maximum Marks: 50 **Dated:** 13, November, 14:30 **Time Allowed:** 80 Minutes

- *All questions are self-explanatory and require no further explanations during exam time.*
- *Make sure that you have signed the attendance sheet before leaving the examination room.*
- *Return the question paper along with the answer sheet.*
- *Attempt all questions.*

(CLO – 1: Employ linear data structures to solve computing problems.)

Q.# 1. Suppose you have BrowingHistory class with data members pageTitle and browsingDate. It also has all the required methods to manipulate its objects in memory. You are required to implement a system for managing internet browsing history. **[7 Marks]**

a. Describe a data structure that you will choose for the given scenario.
b. Write a C++ code to maintain user's internet browsing history in memory, in which users can navigate forward and backward through their browsing history.
c. Write a C++ function to clear the browsing history using choices: last day, last 7 days, last 4 weeks, and all.

Q.# 2. You have studies both queue and stack classes. Suppose you already have an existing class student with data members studentId, name, and CGPA. All the required methods to manipulate Student objects are already built.

Write C++ code for StudentStack class that keeps the records of students in such a way that the student with highest CGPA is always found at the peak of the stack. Just provide the required functionality. There is no need to rewrite existing classes. **[8 Marks]**

(CLO – 2: Use non-linear data structures to solve computing problems.)

Q.# 3. Assume that you have implemented the Doubly Linked List with all its operations. Now you are required to convert that Doubly Linked List (you can change links of next and previous but can't change the addresses of nodes) into a more efficient data structure to perform following operations:

- I could find minimum without traversing all linked list nodes.
- I could find maximum without traversing all linked list nodes.
- I could display all data in ascending order.
- I could find all smaller values of a specific node without traversing all the node.
- I could find all greater values of a specific node without traversing all the node.
- I could perform search in efficient way.