Department of Computer Science, New Campus

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, LAHORE

Paper: Mid_Term **Semester: Fall 2023**

Time Allowed: 40 Minutes **Marks: 20**

Subject: AI Lab

Roll #: _____ **Name: _____**

Implement C++ code.

Question 1: **(CLO1, CLO2)**

Imagine you are developing a simple train reservation system where passengers can book tickets for a train journey. The system needs to manage passenger information and ticket reservations efficiently.

Here are the components:

Linked List for Passenger List: Each passenger is represented by a node in a linked list. The linked list maintains passengers in the order they book their tickets.

Stack for Waiting List: A stack is used to store passengers on a waiting list. Passengers on the waiting list are pushed onto the stack when the train is fully booked.

Queue for Confirmed Bookings: A queue is used to store passengers with confirmed bookings.

Confirmed bookings are enqueued at the end of the queue.

Operations:

Book Ticket:

A new passenger is added to the linked list. If there are available seats, the passenger gets a confirmed booking and is enqueued in the queue. If the train is fully booked, the passenger is added to the waiting list using the stack.

Cancel Booking:

A passenger can cancel their booking. If the canceled passenger was on the waiting list, the passenger is removed from the stack. If the canceled passenger had a confirmed booking, the passenger is dequeued from the queue.

Display Passengers:

Display the details of all passengers, including those on the waiting list and those with confirmed bookings.

Question 2: **(CLO2, CLO3)**

Imagine you are developing a task scheduling system for a computer. Each task has a priority level and estimated execution time. Your goal is to use a combination of a linked list, a stack, and a queue to manage the tasks efficiently.

Here are the components:

Linked List for Task List: Each task is represented by a node in a linked list. The linked list maintains tasks in the order they are added.

Stack for High Priority Tasks: A stack is used to store tasks with high priority. High-priority tasks are pushed onto the stack for quick retrieval.

Queue for Regular Priority Tasks: A queue is used to store tasks with regular priority.

Regular priority tasks are enqueued at the end of the queue.

Operations:

Add Task:

A new task is added to the linked list. If the task has high priority, it is also pushed onto the stack. If the task has regular priority, it is enqueued at the end of the queue.

Execute Task:

The system executes tasks in the following order: Execute a high-priority task from the stack if available. If the stack is empty, execute a regular-priority task from the front of the queue.

Remove Task:

A task can be removed from the system based on its ID.