

Assignment Report: Implementation of Queues Using Double-Linked Linear Lists

Introduction:

The assignment titled "Queues Using Double-Linked Linear Lists" required the implementation of a queue data structure using C++. The main objective was to create a queue that utilizes a double-linked linear list for efficient insertion and deletion operations. The assignment aimed to assess comprehension of data structures, particularly queues, and proficiency in programming concepts such as pointers and memory management.

Code Structure:

The implementation consists of the following components:

1. Node Class (Node.h, Node.cpp):

- Defines a node structure for the double-linked linear list.
- Contains an integer data field (`elem`) and pointers to the previous and next nodes.
- Provides constructors to initialize nodes with and without data.

2. IQueue Class (IQueue.h, IQueue.cpp):

- Represents the queue data structure using a double-linked linear list.
- Supports enqueue, dequeue, peek, size retrieval, and printing of queue elements.
- Implements memory management for dynamic memory allocation and deallocation of nodes.

3. Main Function (IQueueMain.cpp):

- Implements a test scenario to evaluate the functionality of the queue.
- Generates randomized trials to enqueue and dequeue elements from the queue.
- Displays the current size and contents of the queue after each operation.

Key Features:

- Utilizes a double-linked linear list for efficient insertion and deletion operations in the queue.
- Implements error handling to address scenarios such as dequeuing from an empty queue.

- Utilizes randomized trials to comprehensively test the queue's functionality under varying conditions.
- Adheres to standard C++ coding conventions and practices for readability and maintainability.

Possible Improvements:

While the implementation fulfills the requirements of the assignment, potential areas for improvement include:

- Adoption of modern C++ features such as smart pointers for enhanced memory management.
- Enhanced error handling mechanisms using exceptions for better control flow and error reporting.
- Utilization of the `<random>` library for improved random number generation and portability.

Conclusion:

The implementation of "Queues Using Double-Linked Linear Lists" demonstrates a sound understanding of data structures and proficient programming skills in C++. The queue data structure efficiently manages data using a double-linked linear list, providing reliable insertion and deletion operations. Overall, the assignment showcases competency in implementing fundamental data structures and applying programming concepts effectively.