Lab task 9

#include <iostream>

Using namespace std;

// Node structure for the queue

Struct Node {

Int data;

Node\* next;

};

// Queue class

Class Queue {

Private:

Node\* front;

Node\* rear;

Int count; // To keep track of the number of elements

Public:

Queue() : front(nullptr), rear(nullptr), count(0) {}

// Method to add an element to the queue

Void enqueue(int value) {

Node\* newNode = new Node{value, nullptr};

If (rear) {

Rear->next = newNode; // Link new node at the end

} else {

Front = newNode; // If queue is empty, front points to new node

}

Rear = newNode; // Update rear to the new node

Count++; // Increment count

Cout << value << “ enqueued to the queue.” << endl;

}

// Method to remove an element from the queue

Void dequeue() {

If (isEmpty()) {

Cout << “Queue is empty. Cannot dequeue.” << endl;

Return;

}

Node\* temp = front;

Front = front->next; // Move front to the next node

Delete temp; // Free the old front

Count--; // Decrement count

If (front == nullptr) {

Rear = nullptr; // If queue becomes empty, reset rear

}

Cout << “Dequeued from the queue.” << endl;

}

// Method to count the number of elements in the queue

Int size() const {

Return count;

}

// Method to clear the entire queue

Void clear() {

While (!isEmpty()) {

Dequeue(); // Dequeue all elements

}

Cout << “Queue cleared.” << endl;

}

// Method to check if the queue is empty

Bool isEmpty() const {

Return front == nullptr;

}

// Method to display the queue contents

Void display() const {

If (isEmpty()) {

Cout << “Queue is empty.” << endl;

Return;

}

Node\* temp = front;

Cout << “Queue elements: “;

While (temp) {

Cout << temp->data << “ “;

Temp = temp->next;

}

Cout << endl;

}

~Queue() {

Clear(); // Clear the queue in the destructor

}

};

Int main() {

Queue q;

// Enqueue elements

q.enqueue(10);

q.enqueue(20);

q.enqueue(30);

// Display queue contents

q.display();

// Count elements in the queue

Cout << “Number of elements in the queue: “ << q.size() << endl;

// Dequeue an element

q.dequeue();

q.display();

// Clear the queue

q.clear();

q.display();

return 0;

}