**University of Michigan – Dearborn**

**CIS 200 – Computer Science 2**

**Lab# 7**

Quan Le

[lmmquan@umich.edu](mailto:lmmquan@umich.edu)

November 6, 2024

**Table Content**

Contents

[Question 1 3](#_Toc181822978)

[Source Code 3](#_Toc181822979)

[Description 3](#_Toc181822980)

[Structures 3](#_Toc181822981)

[Screenshots 4](#_Toc181822982)

[UML Diagram 5](#_Toc181822983)

[Testing 5](#_Toc181822984)

# Question 1

## Source Code

The source code for this question has been uploaded to Canvas as Lab\_7.cpp.

## Description

* This program creates a DTC tracker using a doubly linked list. Each DTC entry represents an issue detected by the EV’s onboard diagnostics system, allowing users to record, view and remove codes dynamically
* The program offers convenient access to DTCs in either historical or reverse chronological order.It’s especially useful in diagnostics where they can skip through DTCs without having to start from the beginning or end of the list.
* When nodes (DTC entries) are added or removed in a double linked list, the shifting element in more traditional structures is inefficient. The double linked list structure permits previous and next entries to be directly attached to DTCs and, by doing so, also provides support for context sensitive management and tracking of related errors.
* They may start from the historical beginning if a problem occurs several times. These are more likely to be able to focus on the latest DTCs, which may highlight new faults that have arisen, for pressing safety critical issues and diagnostics, as it allows technicians to move through DTCs without needing to restart from the beginning or end of the list.

## Structures

* Define a DTCNode class with string code, description and DTCNode\* next and DTCNode\* prev
* Define a DTClist class with DTCNode\* head and DTCNode\* tail

## Screenshots

Testing case 1

A screenshot of a computer

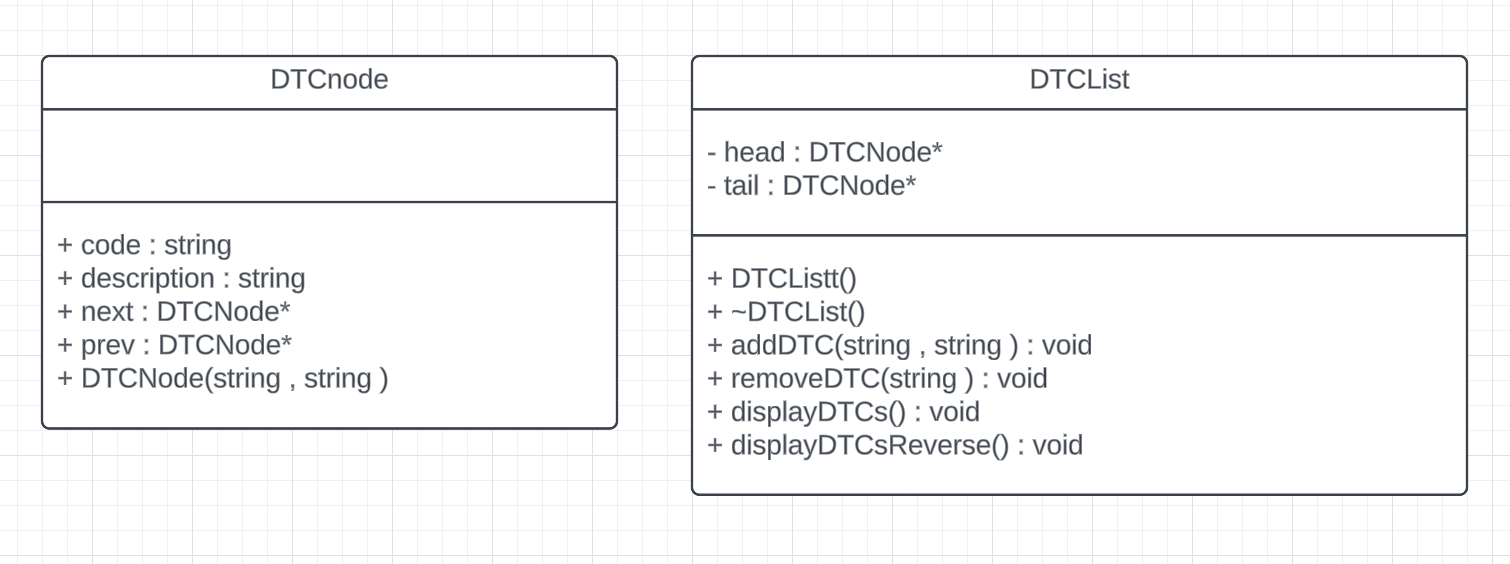
Description automatically generated

Testing case 2

A black screen with a black border

Description automatically generated

# UML Diagram



# Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test # | Input value | Expected output | Actual output | Test Pass/Fail |
| 1 | Insert :  P0A1F, Electric Drive Motor Temperature Too High  P0A80, Replace Hybrid Battery Pack  P0AA6, Hybrid Battery Voltage Isolation Fault  P1A10, Battery Management System Communication Fault  ->Output from oldest to newest  ->Output from newest to oldest  Remove code “P0A80”  ->Output from oldest to newest  ->Output from newest to oldest | Like expected output example in the Lab assignment | Like expected output example in the Lab assignment | Pass |
| 2 | **Don’t input code or description**  ->Output from oldest to newest  ->Output from newest to oldest  Remove code “P0A80”  ->Output from oldest to newest  ->Output from newest to oldest | No DTCs to display  Nothing to remove. Empty list  No DTCs to display | No DTCs to display  Nothing to remove. Empty list  No DTCs to display | Pass |