**Car Showroom Management System - Documentation**

**Overview**

The **Car Showroom Management System** is designed to manage the cars in a showroom using a **linked list** structure. The system allows users to perform several tasks such as adding, searching, deleting cars, processing cars based on the **First Come First Serve (FCFS)** scheduling algorithm, and identifying the cheapest car in the showroom.

**Class Structure: Node**

**Purpose:**

The Node class represents a single car in the showroom and forms the fundamental building block of the linked list. Each Node holds the information about a car and points to the next car in the list.

**Attributes:**

* **model**: Stores the model name of the car.
* **company**: Stores the company name of the car manufacturer.
* **year**: Stores the manufacturing year of the car.
* **price**: Stores the price of the car.
* **next**: A pointer to the next Node in the list, allowing the linked list structure.

**Global Variables**

* **head**: A pointer to the first node of the linked list. It is initialized as nullptr, and if cars are added to the showroom, this pointer will point to the first car.

**Functions**

**1. countCars()**

* **Purpose**: This function counts the total number of cars in the showroom by traversing the entire linked list.
* **Return Value**: An integer representing the total number of cars in the showroom.

**2. addCar()**

* **Purpose**: Adds a new car to the showroom.
* **Process**:
  + Prompts the user to input the car’s **model**, **company**, **year**, and **price**.
  + Validates that the **year** is within a valid range (1950-2025).
  + Appends the new car at the end of the linked list.
  + Prints a success message along with the total number of cars in the showroom.

**3. displayCars()**

* **Purpose**: Displays all the cars currently available in the showroom.
* **Process**:
  + If there are no cars, it prints a message indicating that the showroom is empty.
  + If cars are present, it traverses the linked list and prints each car’s details (model, company, year, price).

**4. searchCar()**

* **Purpose**: Searches for a car by its model.
* **Process**:
  + Prompts the user to enter a car’s **model**.
  + Searches the linked list for the car with the given model.
  + If found, it displays the car's details; if not, it notifies the user that the car is not found.

**5. deleteCar()**

* **Purpose**: Deletes a car from the showroom by its model.
* **Process**:
  + Prompts the user to enter the **model** of the car to delete.
  + Searches for the car in the linked list.
  + If found, it removes the car from the list and deletes the corresponding node.
  + Prints a success message along with the updated count of cars in the showroom.

**6. processCarsFCFS()**

* **Purpose**: Processes the cars in the order they were added (First Come First Serve - FCFS).
* **Process**:
  + Traverses the linked list and prints the details of each car in the order they were inserted.

**7. showCheapestCar()**

* **Purpose**: Displays the cheapest car in the showroom.
* **Process**:
  + Traverses the linked list to find the car with the lowest price.
  + Displays the details of the cheapest car found.

**Main Menu**

The program provides the following options for the user:

1. **Add Car**: Prompts the user to enter details for a new car and adds it to the showroom.
2. **View All Cars**: Displays all the cars currently in the showroom.
3. **Search Car by Model**: Allows the user to search for a car by its model.
4. **Delete Car by Model**: Prompts the user to delete a car from the showroom by its model.
5. **Process Cars (FCFS Scheduling)**: Processes all cars based on First Come First Serve scheduling.
6. **Show Cheapest Car**: Displays the car with the lowest price.
7. **Exit**: Exits the program.

The program continuously shows the menu to the user until they choose to exit.

**Example Interaction**

**User Menu:**

pgsql

CopyEdit

--- Car Showroom Management ---

1. Add Car

2. View All Cars

3. Search Car by Model

4. Delete Car by Model

5. Process Cars (FCFS Scheduling)

6. Show Cheapest Car

7. Exit

**Sample User Actions:**

1. **Add Car**:
   * User enters car details: Model: **Civic**, Company: **Honda**, Year: **2022**, Price: **25000**.
   * The car is added to the showroom.
2. **View All Cars**:
   * Displays the car details: Model: **Civic**, Company: **Honda**, Year: **2022**, Price: **$25000**.
3. **Search Car by Model**:
   * User enters **Civic** as the model, and the car's details are displayed.
4. **Delete Car by Model**:
   * User enters **Civic**, and the car is deleted from the list.

**Conclusion**

This **Car Showroom Management System** efficiently manages cars using a linked list, offering various functionalities like adding, displaying, searching, and deleting cars. It also incorporates features like **First Come First Serve (FCFS)** processing and finding the **cheapest car**, providing an organized and intuitive solution for showroom management.