Program Overview:

The program manages a car rental system for three types of users: Customers, Employees, and the Manager of the store. Each user type has specific operations they can perform.

1.For Customers:

1.1 View Available Cars:
Enter a unique ID.
If the ID is valid, display all available cars in the store.

1.2 Rent a Car:
Enter a unique ID.
If the ID is valid, display available cars.
Choose a car by its ID.
Enter a valid booking date.
Rent the chosen car.
Note: Customers have a record that decreases under certain conditions:
Late returns (after 30 days).
Deteriorated car condition upon return.
Customers have a maximum limit of rented cars based on their record.

1.3 View Due Dates:
Display due dates of rented cars.

1.4 Return a Car:
Enter the car model.
Check if the car is rented to the customer.
Enter the return date and update fines if any.
Update car condition.

1.5 Clear Dues:
Clear previous dues if any.

1.6 View Rented Cars:
Display cars currently rented by the customer.

2.For Employees:
Operations are similar to customers with a 15% discount on rental prices.

3.For the Manager:

3.1 Add a New Car:
Enter car model details if the model ID is not already present.

3.2 Add a New Customer:
Enter customer details if the ID is not already present.

3.3 Add a New Employee:
Enter employee details if the ID is not already present.

3.4 Update Details:
Update car, customer, or employee details based on ID.

3.5 Delete:
Delete a car, customer, or employee based on ID.

3.6 View All Cars:
Display all cars in the store.

3.7 View Rentals:
Display cars rented with corresponding renters.

3.8 Find a Car:
Display details of a specific car based on ID.

- Error Handling:

Inputs are robustly validated:
Invalid dates.
Return date before booking date.
Non-existent IDs (Customer, Employee, Car, Manager).
Invalid input types and ranges.

- Assumptions:

IDs (Car, Customer, Employee, Manager) are unique.
Fine is Rs.2 per day after the due date.
Maximum rented cars for a user is based on their record.
Only one Manager exists.

- Data Integrity:

The system ensures data integrity by verifying the uniqueness of IDs before adding new entries to the database.

- Running Instructions:

-Software Environment:

The software is developed using C++.

No external utilities or database management tools are utilized for storing the database.

Data structures such as maps and vectors are employed for database storage

-Compilation: Use g++ filename.cpp to compile.

-Execution: After the successful compilation of the program use

.\a.exe to execute the program