

## Assignment-1

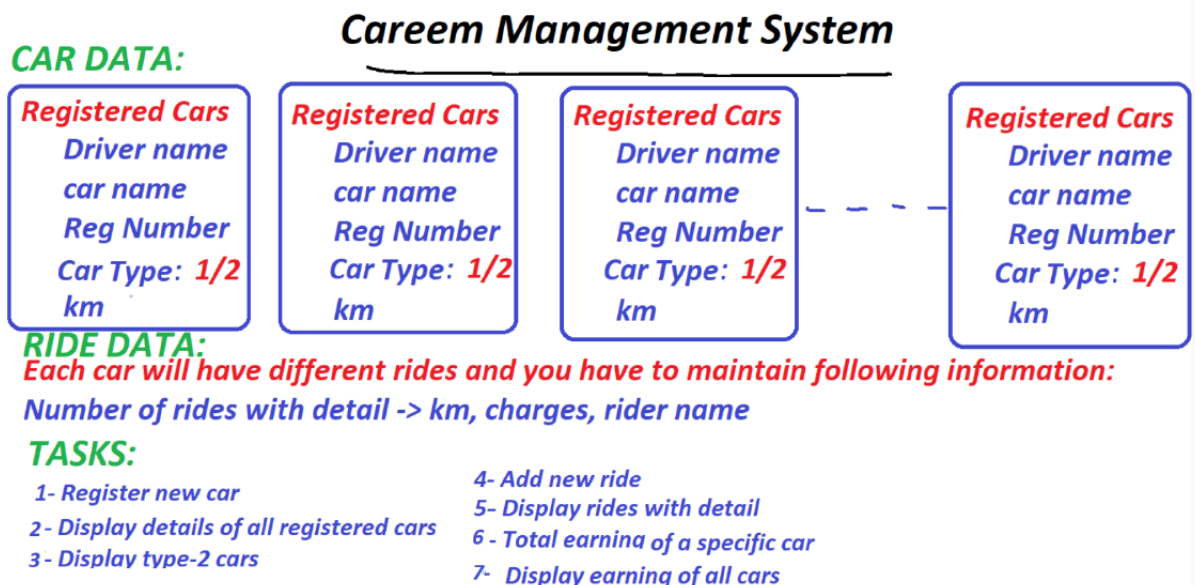
### Singly Linked List Challenges:

- 1- Display the linked list in reverse order by iteration.
- 2- Reverse the linked list.
- 3- Merge Two Linked Lists in Sorted Order.
- 4- Remove repeated values from linked list.
- 5- Find the intersection of two linked lists.
- 6- Delete the nth node from end.
- 7- Swap the node of nth position from start with nth position from end.
- 8- Split the linked list on basis of even and odd numbers.
- 9- Rotate the linked list (swap 1<sup>st</sup> with last node, 2<sup>nd</sup> with 2<sup>nd</sup> last node and so on).
- 10- Implement the concept of **Skip list**.

Consider following points before implementation:

- a. After changing the location of node, the address of node should not change.
- b. You can't use addition array.
- c. You can't count the number the nodes in the linked list.

### Doubly Linked List Challenge:



Hint:

You have to create two different structures of car and rides. Car structure will have the pointer of ride class/structure in-order to access the LinkedList of rides associated with each car.

### **Circular Linked List Challenge:**

Create two circular linked lists (one is of even numbers and other of odd numbers), find and remove minimum odd number from list1 and insert in third list (with same address) named mergelist. In second step find the minimum even number from list2 and merge in list3 named mergelist. Continue the process until both the lists become empty.