List of Cars

This lab combines concepts from several previous labs. It provides practice with linked lists, classes, and overloaded operators.

Program Design

You will create two classes and then use the provided test program to make sure your program works. This means that your class and methods must match the names used in the test program.

You should break your class implementation into multiple files. You should have a car.py that defines the car class and a list.py that defines a link class and the linked list class.

When all is working, you should zip up your complete project and submit it.

Car Class Requirements

Your car class should have the following:

initializer that takes a make, color, and year (default: black 1910 Ford)
setters and **getters** for all three variables

Overloaded equality operator that matches all three variables, return true if they match, false otherwise

Overloaded str() method that returns the contents as (color year make)

List Class Requirements

Your list class should have the following:

a **link class** that contains a pointer to next and a reference to a car object

addCar – input is make, color, year. Creates a new car, creates a new link that points to that car and adds it to the head of the list

findCar – input is make, color, year. Creates a temporary car with those inputs. Uses the overloaded equality operator to check the list to see if such a car exists. Returns true if found, false otherwise

removeHead – if list is empty, returns none. Otherwise returns the car at the head of the list and removes it from the list

Overloaded str() method that uses the car str() method to create a string of all the cars in the list. The cars should be listed one per line.

Overloaded len() method that returns the number of cars in the list