# List of Cars

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

## Program Design

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

Also, you need to break your class implementation into multiple files. You should have a car.h that defines the car class, a list.h, and a list.cpp. The link is a struct that is defined in list.h.

Remember that when writing a method outside of the class, you need to specify the class name in the method name.

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

## Program Requirements

You need to define a class that defines a linked list of cars.

Your car class should have the following:

default constructor that sets up a 1910, Black, Ford
overloaded constructor that takes a string make, string color, and integer year
setters and getters for all three variables

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

Overloaded << operator that adds year, color, make to the provided output stream

Your list class should have the following:

a link struct that contains a pointer to next and a pointer to a car object
constructor that sets head to nullptr
a destructor that walks down the list and deletes any remaining links

addCar – input is make, color, year. Creates a new car, creates a new link that points to that car, adds the link 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 nullptr. Otherwise returns a pointer to the car at the head of the list and deletes that struct

displayList – return a string containing the Cars in the list. Use the overloaded << operator to create this list. There should be a comma and space after each car.

# List of Cars

## Program Hints

Look at the example linked list class for integers provided in Moodle to see how to implement the necessary Linked List methods.

Look at the examples and documentation in Moodle for how to implement overloaded operators for your Car class.

The test driver tests your car class and your list class. Uncomment the TEST_CAR define to test that your car class works properly before you start on the list class. Uncomment the TEST_LIST define to test that your list works.