CARS AND USED CARS

Objectives: Collections, Objects, Inheritance

Task: Hold information about Car inventory using a collection of Car objects, including a UsedCar subclass.

What will the application do?

- Display a set of at least 6 cars (at least 3 new and 3 used)
- Let the user select one of the cars to purchase.
- Print out details of the car they chose, remove it from the list, and print the whole list again.

Build Specifications

- Create a class named Car (5 points) to store the data about a car. This class should contain:
    - Data members for car details
        - A string for the make
        - A string for the model
        - An int for the year
        - A decimal for the price
    - A no-arguments constructor that sets data members to default values (blanks or your choice)
    - A constructor with four arguments matching the order above
    - A ToString() method returning a formatted string with the car details.
        - This method comes from the base object class. Override it.
- Create a subclass of Car named UsedCar (3 points). UsedCar should contain:
    - Data member for used car details:
        - A double for mileage.
    - Constructor: Takes five arguments and calls the four-argument constructor for Car and saves the mileage argument
    - ToString: Returns a formatted string with the used car details
        - This method comes from the base object class. Override it.

- Create an instance of List<Car> that can hold instances of Car and any class derived from Car. **Make this list a *public static* member of Car**.
    - In your main, create at least three Car instances and at least three UsedCar instances and add these six instances to the list
    - Add a public static method to Car called **ListCars** that loops through the list and prints out each member and its index in the list. (Hint: Use a regular for loop, not a foreach loop so you can print out the index.)
    - Add a public static method to Car called **Remove** which takes an integer parameter and removes the car whose index is that parameter
- In your main, print out the list (by calling the ListCar method). Then ask the user which car they would like to buy, by number (the index of the car).
- Print out the details for the chosen car. (Think about how to print out this information: You'll access the item in the list by index, and call Console.WriteLine.)
- Remove the chosen car from the list
- List all the cars again

Hints:

- Use the right access modifiers (public/private/protected)!
- You can just use \t tab escape characters to line things up, or if you want to get fancier, look up text formatters.

Extra Challenges:

- Think about other methods which might be useful for your Car such as "BuyBack" where you can add a used car to the list. Implement them and modify your app to take advantage of them.
- Create an Admin mode which lets the user edit cars.
- Provide search features:
    - View all cars of an entered make.
    - View all cars of an entered year.
    - View all cars of an entered price or less.
    - View only used cars or view only new cars.

See next page for Console Preview.

Console Preview

Your output will vary based on decisions you make with your partner.

```
Welcome to Grant Chirpus' Used Car Emporium!

1. Nikolai Model S    2017       $54,999.90
2. Fourd   Escapade   2017       $31,999.90
3. Chewie  Vette      2017       $44,989.95
4. Hyonda  Prior      2015       $14,795.50 (Used) 35,987.6 miles
5. GC      Chirpus    2013       $8,500.00 (Used) 12,345.0 miles
6. GC      Witherell  2016       $14,450.00 (Used) 3,500.3 miles
Which car would you like?  {6}
GC         Witherell  2016       $14,450.00 (Used) 3,500.3 miles
Excellent!  Our finance department will be in touch shortly.
Have a great day!
```