Car

1. Create a new class that represents a *Car*.
2. Add a *year*, *make*, *age*, and *is classic car* property to the Car class:
    - `year`: indicates the year that the car was manufactured.
    - `make`: indicates the make of the car.
    - `is classic car`: indicates if the car is a classic car.
    - `age`: indicates the age (in years) of the car from the current year.
3. Create a constructor that accepts `year`, `make`, and `is classic car`.
4. Instantiate an object, or objects, in `Main()` and use the object(s) to test your methods.
5. Create a method on Car to determine if a car must receive an e-check using an input parameter: `int yearToCheck`.
    - Return true under the following conditions:
        - Even-model year vehicles must be tested if `yearToCheck` is even.
        - Odd-model year vehicles must be tested if `yearToCheck` is odd.
    - Return false if an e-check is not needed or the car is exempt:
        - A vehicle is exempt if it is under 4 years of age.
        - A vehicle is exempt if it is over 25 years of age.
        - Classic cars are always exempt.
6. Override the `ToString()` method and have it return `"CAR - {year} - {make}"` where `{year}`, `{make}` are placeholders for the actual values. The values from the object should be shown in the string where `{variable-name}` is indicated.
7. Implement unit tests to validate the functionality of:
    - The age calculation
    - The e-check method
8. In the Program class, within the Main method, read in the provided csv file `CarInput.csv`, and use it to populate a list of *Car* objects.
9. Add up the age for all of the cars in the list, and print it to the screen.