**CarFactory Documentation**

**By Chris Ricchi**

**Introduction**

Welcome to the CarFactory program, a trial where our simulated car factory will produce random types of vehicles with their own parameters (type, color, year, miles) and write the data to a .csv files for further evaluation. This program can also import a .csv file of similarly formatted data for evaluation.

**Using BirthdayChecker**

This program functions with the use of four classes: **Car, Factory, CarImporter, and CarExporter**. The first class, **Car**, is used to create a Car object that contains its own type, year, color, and miles. The next class, **Factory**, allows us to create a random Car object with types, years, colors, and miles that make sense in the context of the simulation. The type and color are chosen from an array of strings, the year is a randomly chosen value between an initial date of 1960 and a maximum date of 2023. For more clear distinction in our results, we adjusted the chance of the type of the car being a sedan to 30%. This means that for every car created, there is a 30% chance of the type being a sedan, and a 70% chance of the type being any other possibility. Lastly, the miles on the car is a randomly generated number from 1 to 250,000. The **CarExporter** class loops through 1,000 iterations and creates a new random car from the Factory class per iteration. This data is then written in column-based format and exported to a local file named cars.csv. The **CarImporter** class searches for a local file named cars.csv in the directory of the program and loads the data from each line into a new Car object with the correct data extracted from the cars.csv file. Theoretically, you could export using the CarExporter class and then immediately import the same data again using the CarImporter class.

**Understanding the Results**

By default, the CarFactory simulation will run 1,000 total iterations and generate a new car with randomly chosen attributes per car. When our data is exported to a .csv file, this file can be opened in Microsoft Excel for quick data representation.

**In-Depth Results**