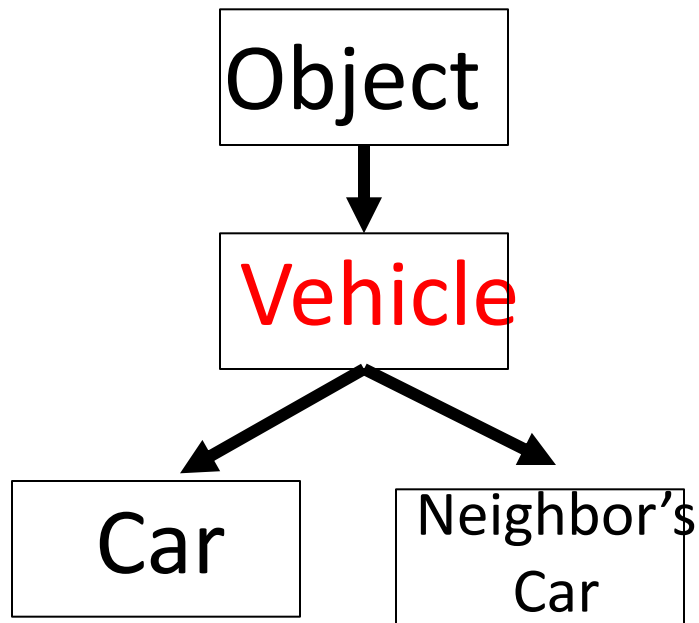# New Idea – Abstract Class

# Mr. Neat
# Java

# Big Idea in Java…
# Inheritance

- Mechanism to make a group of different objects that have something in common
- Mechanism to make a more specialized class from a more general class (eg, Car with person in it)
- The Object class is the mother of all classes
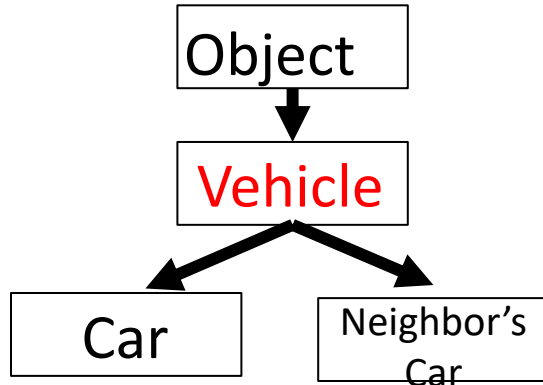- Family trees help to understand

# Big Idea in Java…
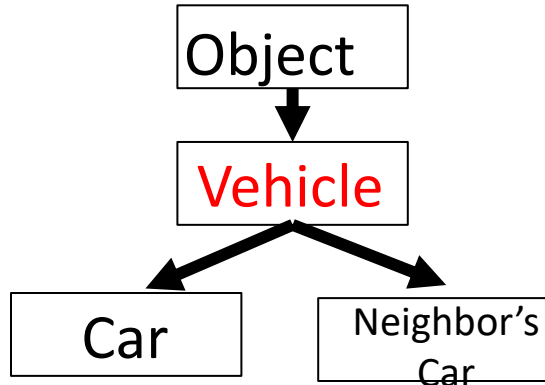## Inheritance – Abstract class

# Big Idea in Java…
## Inheritance – Abstract class

```
┌────────┐
│ Object │
└────────┘
     ↓
┌─────────┐
│ Vehicle │
└─────────┘
   ↙      ↘
┌─────┐   ┌───────────┐
│ Car │   │ Neighbor's │
└─────┘   │    Car     │
          └───────────┘
```

- Make a new file called Vehicle
- Add abstract methods (methods Vehicle will use)
- Extend Cars and Neighbor's Car
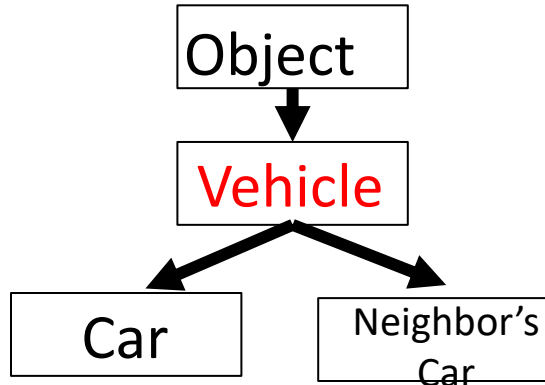
# Big Idea in Java…
# Inheritance – Abstract class

Object

↓

Vehicle

↙ ↘

Car     Neighbor's Car

- Make a new file called Vehicle
- Add abstract methods (methods Vehicle will use)
- Extend Cars and Neighbor's Car

public abstract class Vehicle
{


}

# Big Idea in Java…
# Inheritance – Abstract class

| Object |
| --- |

↓

| Vehicle |
| --- |

| Car | | Neighbor's Car |
| --- | --- | --- |

- Make a new file called Vehicle
- Add abstract methods (methods Vehicle will use)
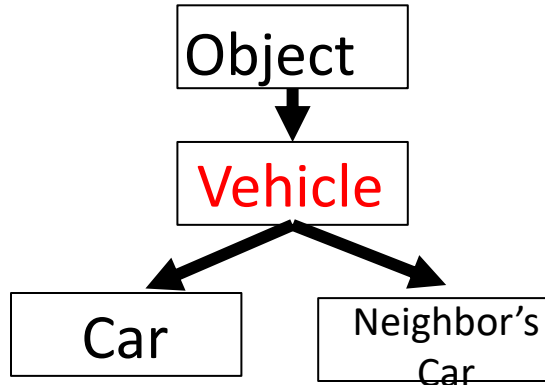- Extend Cars and Neighbor's Car

```
public abstract class Vehicle
{
    public abstract void driveRandom();
    public abstract void draw();
}
```

# Big Idea in Java…
# Inheritance – Abstract class

| Object |
|--------|

↓

| Vehicle |
|---------|

↓ ↘

| Car |  | Neighbor's Car |

- Make a new file called Vehicle
- Add abstract methods (methods Vehicle will use)
- Extend Cars and Neighbor's Car

public class JoeCar extends Vehicle
{

      …

# Lab

- Make 100 Cars made up of instantiations your Car class and your neighbor's
- recycle all of the cars
- Make each car have:
  - Random colors
  - Random names
  - Random location
  - Random speeds

# hints

- Make an array of 100 <span style="color:red">Vehicles</span>
- Fill array (new YourCar(), new JoeCar())
- Recycle them