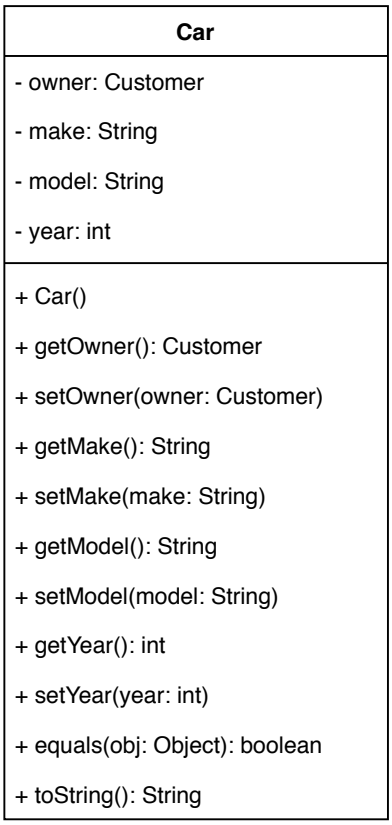
Model 1 Class Diagram

Recall that a UML class diagram summarizes the attributes and methods of a class. In the repair shop example, a `Car` class might look something like this:



Questions (15 min)

Start time:

1. How many _____ are in the diagram above?

a) attributes 4

b) constructors 1

c) getters 4

d) setters 4

2. What is the type of the `owner` attribute? Is it a primitive or reference type?

It is a `Customer`, or more specifically, a reference to a `Customer` object.

3. Explain how this design allows multiple cars to be owned by the same customer.

Multiple `Car` objects can reference the same `Customer` object in memory.

4. List three attributes that would be appropriate for the `Customer` class.

Variable Name	Data Type	Example Value
name	String	"Jonathan Alger"
address	String	"800 S Main St ..."
phone	String	"(540) 568-6211"

5. Rewrite the attributes from the previous question in UML format.

- name: String
- address: String
- phone: String

6. For each attribute, define a `get` method. Write your answer in UML format.

+ getName(): String
+ getAddress(): String
+ getModel(): String

7. For each attribute, define a `set` method. Write your answer in UML format.

+ setName(String name)
+ setAddress(String address)
+ setModel(String model)

Optional Questions

8. What rules might be implemented in the `set` methods to ensure that only valid attribute values are stored? *Example: The customer's name should contain only letters and spaces.*

Variable Name	Validation Rules
name	no special characters, max length (e.g., of 50)
address	use an API like Google Maps to ensure it exists
phone	must be 10-digit number; use parens and dashes

9. Based on the attributes you defined, how could you determine whether two `Customer` objects represent the same customer?

Answers will vary. It's probably good enough to compare the name and phone number. But that would require names and numbers to be stored in a consistent format. A more reliable approach would be to define a unique customer number.

10. In Model 1, what is the parameter name and type for the `equals` method? What version of `equals` is this method overriding?

The name is `obj`, and the type is `Object`. The `Car` class is overriding the default version of `equals` defined in the `Ojbect` class.