**Classes and Objects**

In these exercises you will need to create classes and define their properties and behaviours. Each class should be in its own file. If you need to call any of the methods you define you can do it from the main method inside the Main class.

You should create a new project in IntelliJ for this - remember to create a Git repository too!

**Example** Define a class to describe a pet cat with a name property. Write a getter for name and use it to print the value in Main.

Cat.java

public class Cat{  
  
 private String name;  
  
 public Cat(String name){  
 this.name = name  
 }  
  
 public String getName(){  
 return this.name  
 }  
}

Main.java

public class Main {  
 public static void main(String[] args){  
 Cat cat = new Cat("Star");  
 String catName = cat.getName();  
 System.out.println(catname);  
 }  
}

**Exercise 1**[**​**](https://www.textbook.amigoscode.com/docs/java/exercises/classes-1/#exercise-1)

Create a class to represent a car. The car should have properties representing the manufacturer, price, and engine type (eg. petrol, electric, etc).

**Exercise 2**[**​**](https://www.textbook.amigoscode.com/docs/java/exercises/classes-1/#exercise-2)

Write getters for each of these properties. Test them in Main.

**Exercise 3**[**​**](https://www.textbook.amigoscode.com/docs/java/exercises/classes-1/#exercise-3)

Write setters for the properties. Test them in Main.

**Exercise 4 (optional - slightly harder)**[**​**](https://www.textbook.amigoscode.com/docs/java/exercises/classes-1/#exercise-4-optional---slightly-harder)

Modify the engine type property to use an EngineType enum. You'll need to modify the getter and setter to reflect the different data type.

**Exercise 5**[**​**](https://www.textbook.amigoscode.com/docs/java/exercises/classes-1/#exercise-5)

Create a class to represent a car dealership. It should have properties representing its name, the maximum number of cars it can have on display and all the cars currently in stock.

TIP

We could create an individual property for each car we have in stock, but is there a data type we have already seen which would let us store all our cars together in the same variable? How can we figure out how big that data structure should be?

TIP

A brand new dealership won't have any cars yet. Do we need to pass that information in the constructor when we create a dealership, or can we define our class in such a way that every new dealership starts with no cars?

**Exercise 6**[**​**](https://www.textbook.amigoscode.com/docs/java/exercises/classes-1/#exercise-6)

Write a method to count the number of cars in the dealership. It doesn't need to take any parameters and should return an int. Write another method to add a car to the dealership. The method should have a Car parameter and doesn't need to return anything. Test your method by counting the number of cars, adding a new car then counting again to make sure the number has gone up by one.

Hint: You will need some way of finding the first empty slot in the array to save the Car in. Maybe checking each slot in the array in turn would help?

**Exercise 7 (harder)**[**​**](https://www.textbook.amigoscode.com/docs/java/exercises/classes-1/#exercise-7-harder)

Write a method for the dealership called findCarByManufacturer which takes a String as a parameter and returns a Car object. The method should return the first Car the dealership has in stock with a manufacturer property matching the value passed to the method.

TIP

**Hint 1**: Think carefully about how you will access the information you need from the Cars. Remember the methods you wrote in exercise 2...

TIP

**Hint 2**: To test this in Main you will need to: - create some Car objects - add them to the Dealership - store the result of your findCarByManufacturer method to a variable - print the variable