Challenge 3: Playing with Cars and Engines!

In this exercise, you have to perform composition between the Toyota class and its Engine!

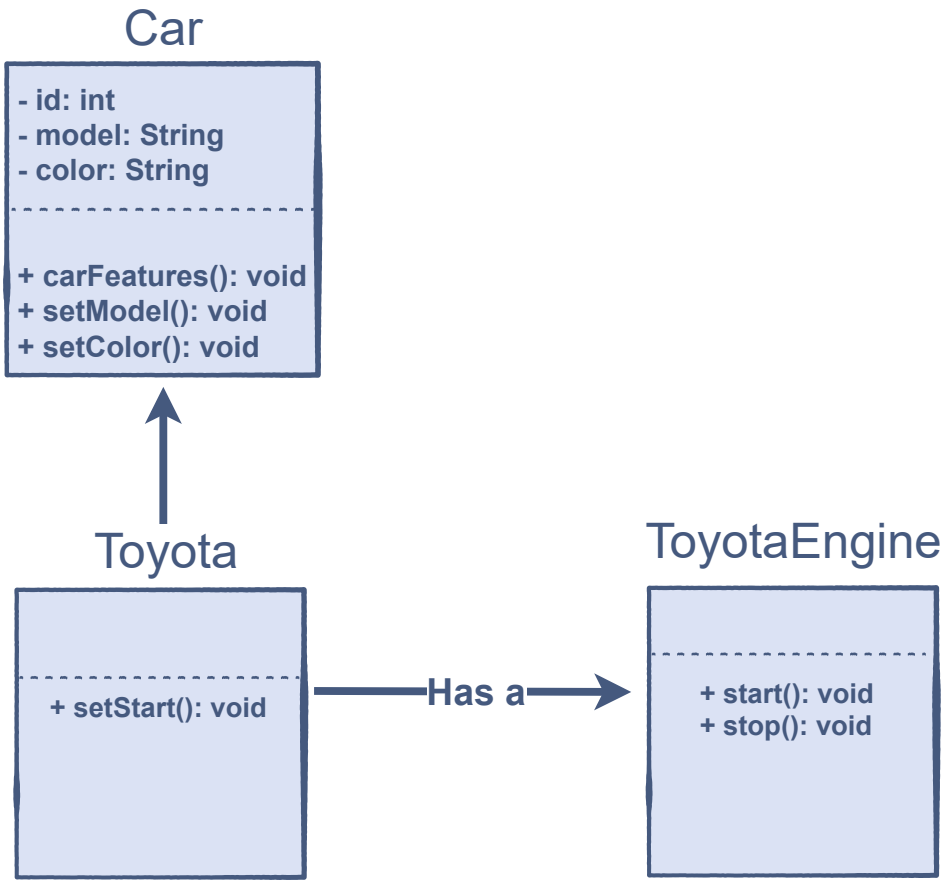
WE'LL COVER THE FOLLOWING

Problem Statement

You have to create a **Toyota** class which inherits from a **Car** class, and is composed of an **Engine**

> **Note:** You already know that in such a composition relation, the *Toyota* class will be responsible for *Engine*'s lifetime.

Consider this diagram for reference:



Car, Toyota and ToyotaEngine: Class Representation

You should implement the `Toyota` class as the **child class** of the `Car`. It should have an instance of `ToyotaEngine` and a `setStart()` function which in turn calls the `start()` function of the `ToyotaEngine` class.

The `start()` and `stop()` functions of `ToyotaEngine` class simply print out on screen that the engine has been started/stopped.

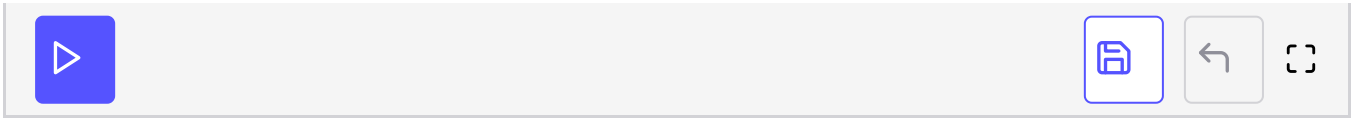
Coding Exercise

First, take a close look and design a step-by-step algorithm before jumping to the implementation. This problem is designed for your practice, so initially try to solve it on your own. If you get stuck, you can always refer to the solution provided in the solution review.

Good Luck!

```
class Car {

  // Complete its implementation

}
```

```
}

class Toyota extends Car {


  // Complete its implementation

}

class ToyotaEngine {

  // Complete its implementation

}

class Main {

  public static void main(String[] args) {
    // Write your code here
  }

}
```



The solution is explained in the next lesson!