This assignment builds upon the work we've done in lectures so far.

A company called  ‘MkX’ cars have designed a revolutionary new modular car. You can design the car online and choose a Car body, a particular Engine, Door, and Wheel types. MkX have designed a simple software simulation in Java to demonstrate how different car/engine/door/wheel configurations work.

Your assignment is to demonstrate this simulation.

For example, to demonstrate the Mark7/PL9/UP2/Wichelin8 configuration, you might write the following code in the Code Pad (assuming your classes are coded correctly):

Car car = new Car("Mark7");

Engine engine = new Engine("PL9", 9);

car.add(engine);

Door door = new Door ("UP2", 2);

car.add(door);

Wheel wheel = new Wheel ("Wichelin8", 8);

car.add(wheel);

car.setFuel(100);

car.drive();

car.getDistance();

This program should output how far a particular Car configuration can travel given a full tank of fuel (say 50 units)

This is not as difficult as it might seem.

**1.** **As always,  let's first consider the classes we need:**

* Car
* Engine
* Door
* Wheel

 You can write stub code for these or simply adapts the code from the lecture sessions.

**2. Now let us consider the properties (fields)  of each class**

* The Wheel has a radius and name ("Wichelin8")
* The Engine has fuel level, a name and a fuel efficiency rating  - but in this case it is not kpg (kilometers per gallon) as in the lecture - but wheel turns per litre (the units don't matters)
* The Door has name and the quantity (number of doors) each car have. (For example in this case it has 2 Doors that opens upward (“UP2”).
* The Car also has a name ("Mark7"), has the distance required to be travelled, and the totalNumber of kilometers completed

**3. Now let's consider the the relationships between the Car, Engine, Door and Wheel.**

A Car has an Engine and Door. And an Engine has a Wheel. These are composition relationships. Do you recall from the lecture how composition relationships were coded?

(While automobiles generally have four wheels, we are going to use a single Wheel in this simplified model. Let's consider it to be one of the front two drive wheels. The other is identical so we won't bother about modelling it.)

Given a tank of 50 litres of fuel, each Car can call on the Engine to turn the Wheel a certain number of times. The number of turns will be based on the Engine's **wheel turns per litre**rating.

For each full turn, a wheel will travel the length of its circumference. If you remember your geometry, you will recall that the circumference of a circle is 2π \* r , where r is the radius of the circle.

This means that larger wheels in our simulation will travel further as they have a larger radius.

In Java you can calculate this as

double circumference = 2 \* Math.PI \* radius;

You should have the variable radius declared and initialised before using this piece of code.

The important thing to recall is that Engine does not know the radius of the Wheel, or the distance traveled. Your Wheel object should have a turn() method that returns the distance travelled (circumference) on every turn.

**A key challenge for you in this assignment will be to decide on the methods that belong to each object and what each method should do/return.  Here are some questions you can ask yourself:**

* What value, if any, does the method return?
* Does the method functionality make sense in terms of the information that is held by the object? It should do.
* In order to deliver its own functionality does the method need to call other method(s)  from other object(s)  ?