Lesson 2: Classy Cars (see what I did there?)

What are we going to learn?

What classes and objects are and how to use them.
How to make methods on classes and call them

Why are we learning this?

Classes and objects are a key part of programming. Using classes we can reuse code and make lots and lots of objects very quickly.
Objects are basic building blocks of programming.

Classes

Look at the new code given to you in the example program **carExample**.

Run it! What happens?

Break down of the code:

Here for the first time we have made a **class.** Our class is called Car.
Classes have definitions, we make instances of classes - called objects.

Making a Car object:

We need a way to setup the Car when its starts. Like the `setup` function, but for each object.
This is run for each object when we make a new version of it.
Where it's defined - in the **Car** class.

```
public Car(int inStartHozPos, int inStartVertPoz) {
    carWidth = 80;
    carHeight = 40;
    speed = 2;
    carHozPoz = inStartHozPos;
    carVertPoz = inStartVertPoz;
    redAmount = 255;
  }
```

This happens when we make a new Car, the same way we make a new `List`. We provide parameters - like when we call `ellipse`. Using the following code:

```
Car newCar = new Car(20, 60);
```

Tasks to get used to constructors and making new cars:

Change the code so that:

- The car starts at a different position. (This code **is not** in the Car class)
    - Change the starting horizontal position
    - Change the starting vertical position

    - When the car moves - where does it move from? Does that change? Why?
- Change the size of the car. (This code **is** in the Car class)
- Change the `speed` of the car - what does this affect? (This code **is** in the Car class) Where is the `speed` variable used in the Car class?
- Make a second Car object. You need:
    - A new car variable.
    - To call `.drawCar()` on that in the **`draw`** method.
        - What happens when you don't do that?
    - Is the speed of this second Car the same as the first car? What about the size?

Objects have variables and functions.

Just like in "normal" programing objects have functions and variables.
Objects have **variables** and **functions**. They can **hold data** and **do things**.

But each object has a different set of variables belonging to it.
And calling a function on a object does that function on that object.

You can think of yourself, and everyone in the class as an object:

- You have **variables** (bits of data) eg your: name, age, place of birth etc.These are unique to you. This is **your** data.
- And functions, so you can do stuff. Like wake up, run, walk eat etc. This functions are done by **you.**

Tasks to get used to variables and functions on cars:

We've already changed one variable on the cars - the speed. And we saw how it was used.

- At the moment our Car class looks a bit weird - there's **only one wheel**! Lets change that! Go into the `drawCarImage()` method in the car class and add code to draw the other wheel. Does this affect all the cars? Why?
- Change the speed of the `exampleCar` object. See the `setSpeed` method in the Car class. It changes the speed variable of the Car. So lets do that in the **`void setup`** method.

```
exampleCar.setSpeed(5);
```

- What happens now? Does the speed of the second Car change?
- Make **new function** in the Car class! This method will be very like the `setSpeed`, but will be called `setRedAmount`. It will change the `redAmount`. It will take in one number. Before you run it - what do you think it will do?
    - If you feel like it add two more variables: `greenAmount` and `blueAmout`.
    - Change `setRedAmount` to `setCarColour` and now take 3 numbers!
    - Don't forget to use `greenAmount` and `blueAmount` in a `fill` function!
- Lets make a **proper race app**!
    - Use the `setCarSize` method to change the size of each of the Cars. So they both look different. Like we did above with the speed.
    - Make 3rd and 4th Car objects.

    - For each car set the **speed to a random number** between 1 and 5.
    - Run the app what happens? Does this happen every time?
- Check when a car has finished.
    - check each of the Cars and see if the `carHozPoz` for that car is over the edge of the screen. If so finish the race!
        - Extras:
            - Show an end game screen if a car has won!
            - Get the number of the car that finished first!
            - Add custom image to the background of the app!
            - Make more improvements to the `drawCarImage()` to draw a fancier car. Do you remember how to make curved edges on rectangles? Look up rect in the reference guide for hints!