Static Methods

In this lesson, we will discuss static methods.

We'll cover the following

Introduction to static methods

The keyword `static` is a useful tool in Java. It basically means that a **static** method is one that can be called without any object instance. There are a few key points to remember with **static methods**.

- These methods **do not** belong to any particular object but rather to the whole class.
- This method can be invoked **without creating** an object instance.
- Keep in mind that **non-static** data members cannot be used and neither can **non-static** methods be called directly in *static methods*.

To understand by example, we look at the `System.out.println()` method. This method is a static method of the `System` class. Making this method static has its own benefit- you can use **one** output stream for **all** printing of outputs.

With that, let's look at another example of a static method in the snippet below.

```
class car {

    public String colour;
    public car() {};
    public car(String col) {
        this.colour = col;
    }
    public String getColour() {
        return this.colour;
    }
    public static void main(String[] args) {
        car newCar = new car("Pink");
        System.out.print("Car colour is: " + newCar.getColour() + "\n");
```

```
        setColourBlue(newCar);
        System.out.print("Car colour is: " + newCar.getColour() + "\n");
    }

    public static void setColourBlue(car c) {
        c.colour = "Blue";
    }
}
```

In the above code `setColorBlue()` is a static method. That's why on *line 14* we called this method without creating an object.

`main` is static method

See the code given below!

```
class static_methods {
    public static void main(String[] args) {
        int my_variable = 0;
        System.out.println("Printing variable: " + my_variable);
    }
}
```

Note: main() is a Static method

In the code above, `main()` is a static method. Why does it **need** to be static? This method is called even when no object is created in the code. It does not need an object to be present for its functioning. Hence, this method has the `static` keyword attached to it.

Well, this is all about **methods**. Next up, some challenges are provided to test the understanding of methods in Java.