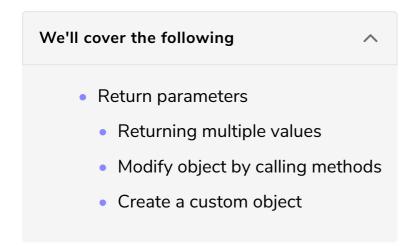
#### **Return Parameters in Methods**

In this lesson, we will discuss return parameters in methods.



# Return parameters #

*Return parameters* are ones that the method will **send back** to the class from where it was called. To understand, let's look at the diagram below which shows a function that changes a *String* value and returns the new **modified** String.

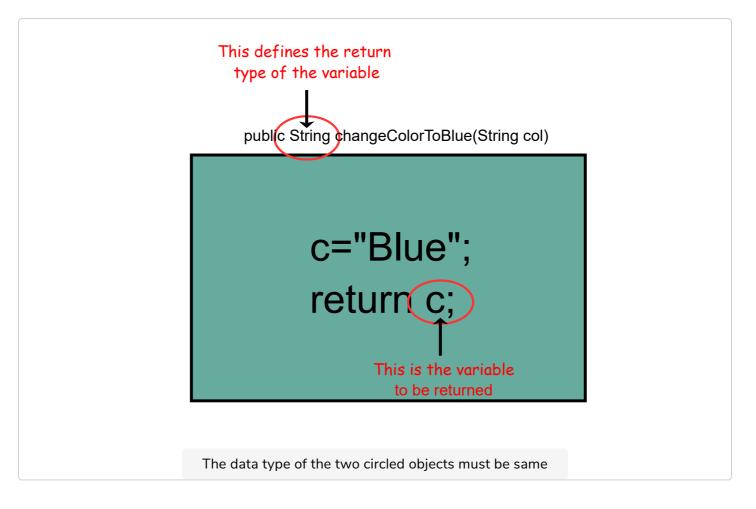
```
class car {
    public String colour;
    public car() {};
    public car(String col) {
        this.colour = col;
    public String getColour() {
        return this.colour;
    public void changeColourToBlue() {
           this.colour = "Blue";
    public static void main(String[] args) {
    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











Another interesting aspect is that a method *may not* return anything. In this case, the **return type** in the method declaration is written as **void**. This is seen in the **car class** example above in the method that changes car colour to "Blue" (line 11-13).

## Returning multiple values #

A method can only return one value. However, if the user wants to return multiple values, then two ways can be adopted.

#### Modify object by calling methods #

The first way is by defining a class and calling methods that modify the objects themselves.

## Create a custom object #

Another way is to create *custom* objects which means that you specify a special object that can store multiple types of data or values, as per requirement and **return** that!

In the next lesson, we will discuss constructors in detail.