Methods in Java

In this lesson, the basics of Methods in Java are explained.

We'll cover the following

Methods are how we **communicate** with objects in Java. We *invoke* or *call* a method, i.e.; we are asking the object to carry out a task *through* the method. Hence, a **method** is a mechanism that allows us to *implement functionality* or *attribute behavior* to objects. Methods follow a particular syntax, as shown in the snippet below.

```
ClassOne{

  visibility ReturnType Methodname(parameter_one, parameter_two){

    //method body

    return returnVariable;
  }
}
```

Understanding syntax

Visibility

Visibility is the scope of a Method, i.e., how accessible it is from different places in the programs. There are **four types** within visibility:

- public: The method can be called *anywhere* in the application.

For now, we will consider only public methods, and later on, when we do understand classes and packages, we'll talk about other options.

ReturnType

This defines whether the method **returns** a value or not. If it does, then this is where the **data type** of the returning variable is mentioned in the declaration. If the method *does not return* anything then the type is set to **void**. Otherwise, it can return **primitive** as well as special data types.

Methodname

This is where you define the name for your method. Like a variable identifier, this name **should** be descriptive of the functionality of the method. For instance, the **substring** method for Strings that we have learned in this lesson. This method has an explanatory name that identifies what this method does. Imagine if it had a vague name like **partOfString** or **smallString** which could be interpreted to multiple meanings, it would be so confusing!

Parameters

The ‘Methodname’ is followed by two round brackets, `()`. Within these round brackets, the parameters are passed to the method. These parameters are used within the method body, which is enclosed in curly brackets, `{}`. Parameters within the brackets `()` are separated by commas and can have multiple data types.

returnVariable

The ‘returnVariable’ is the variable whose value is to be returned outside the method body. This variable must have the *data type* of the returnType signified in the method declaration line.

Let’s look in detail at parameters being passed to a method in the next lesson.