**Java toString method**

**[ Note:** If you just wanna learn how to use **toString()** then start reading from page 2, page 1 focuses on the story behind **toString()** **]**

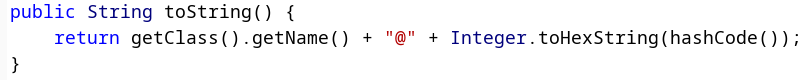
We all know by now what actually happens when we try to print an Object or an Array in Java. Let’s see four examples of this.

|  |  |
| --- | --- |
|  | |

As we can see in the four examples above, the output specifies which type of Array/Object it is and some random numbers. I for integer, D for double, [Ljava.labg.String for String then @ symbol random alphabets and number.

It is basically a String which is being returned from a method called toString(). But we get this output without ever writing any toString() method, right? So, where is this method actually written?

Hopefully, by now we all know about the **Object Class** which is the parent of all the classes in Java. Starting from Wrapper classes such as Integer class, Double class to any user defined Classes; all of them are implicitly extended from the Object class. I think now you can guess that the mysterious **toString()** method belongs to this **Object class**. That is why, whenever we’re trying to print an object, java looks for the **toString()** method and finds it in the Super Object Class. Let’s take a look inside this default **toString()** method.



**format of the returned String :**

**<fully qualified class name>@<hash code in hexadecimal format>**

What you see above is the default **toString()** method written inside of the **Object class**. Anyway, let’s not dive too deep into this right now. Instead, we can focus on how we can use this in our day to day coding. The fun thing about this method is that we can easily **override** this method to print whatever we want when the **object** is being printed.

As explained in the previous page, we can override the **toString()** method in any class we write. We just have to remember two things about the toString**()** method; firstly it has to be **public** (why?? [**Read this**](https://stackoverflow.com/questions/15467496/why-my-overriding-method-of-tostring-has-to-be-public)), secondly, it must **return a String**. The returned string should concatenate the relevant variables of the class in a String. Let’s see an example:

|  | **OUTPUT:**  Player1: Zeus, Player2: Hades |
| --- | --- |

It’s as simple as that. Basically, now that the original **toString()** method is overridden by our own, so when we print the object we get whatever is returned by the **toString()**. Btw, you can also call the toString method on its own as well; like **g1.toString().** It would give the same output. That’s it. :)