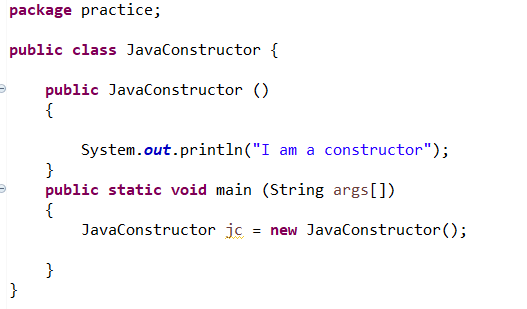
**Constructors in JAVA**

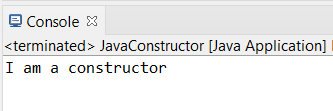
A ***constructor*in Java is a block of code** similar to a method that’s **called when an instance of an object is created**. Here are the key differences between a constructor and a method

* A constructor **doesn’t have a return type**.
* The name of the constructor **must be the same as the name** **of the class**.
* Unlike methods, constructors are not considered members of a class.
* A constructor is called **automatically when a new instance of an object is created.**

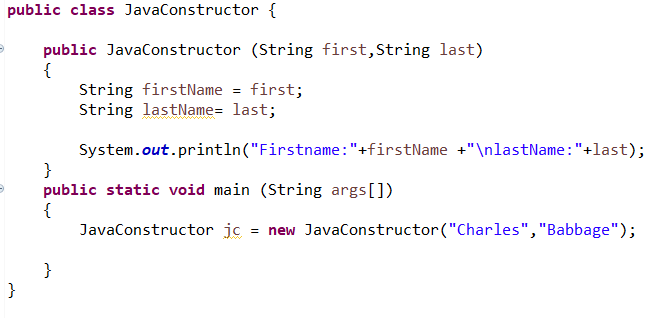
**Constructor implementation**



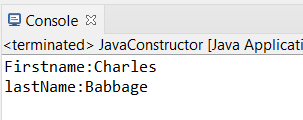
**Result of the above constructor implementation**



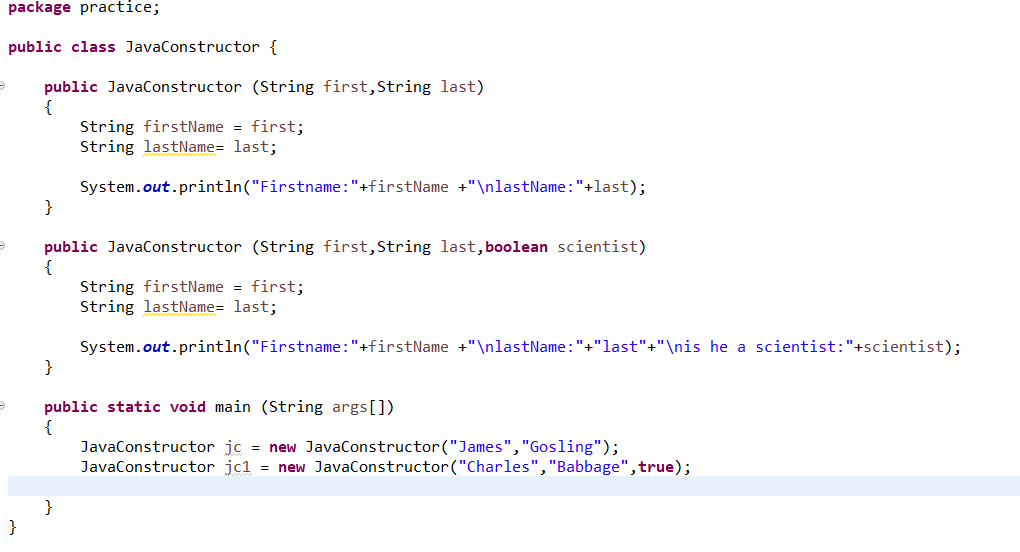
In addition to that **we can code the parameter** list the same way we can code for method.



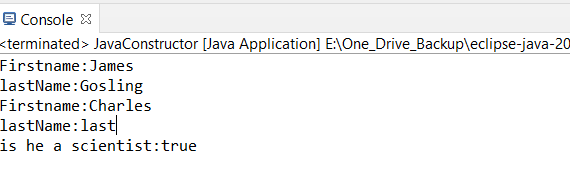
**Result of the above constructor implementation**



Like methods, constructor can be **overloaded, meaning we can create more than one constructor** for a class if each constructor has unique signature(parameter)



**Result of the above constructor implementation**



**Important points to remember while using constructor in java:**

1. When we are not providing a constructor for a class, Java will automatically create a default constructor that has no parameter and its not initialize any fields. This default constructor is called when we specify the new keyword without passing any parameters.



1. When we **explicitly declare any constructors for a class, Java does not create a default constructor for the class**. As a result, if you declare a constructor that accepts parameters and still want to have an empty constructor (with no parameters and no body), you must explicitly declare an empty constructor for the class.