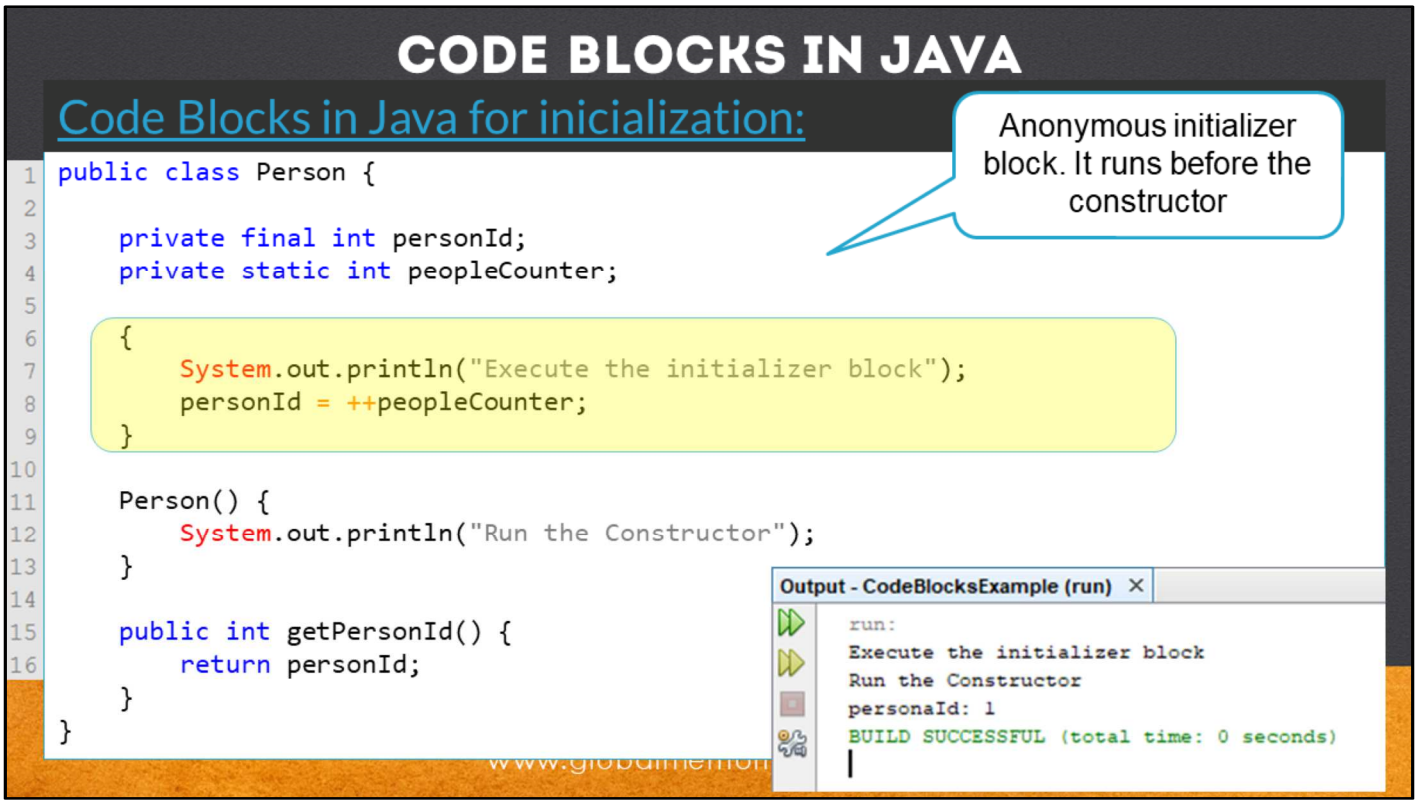
Hello, Ubaldo Acosta greets you again. I hope you're ready to start with this lesson ..

We are going to study the topic of code blocks in Java.

Are you ready? Come on!



A block of code in Java is basically any part of code that starts with an open curly bracket "{" and ends with a close curly bracket "}".

Any content within these curly brackets is known as a block of code. However, there are blocks of anonymous code, and unlike the blocks we have used for example to create a class or a method, anonymous blocks allow us to commonly initialize variables of the class. These blocks called initializers can initiate two types of variables, either instance or static type.

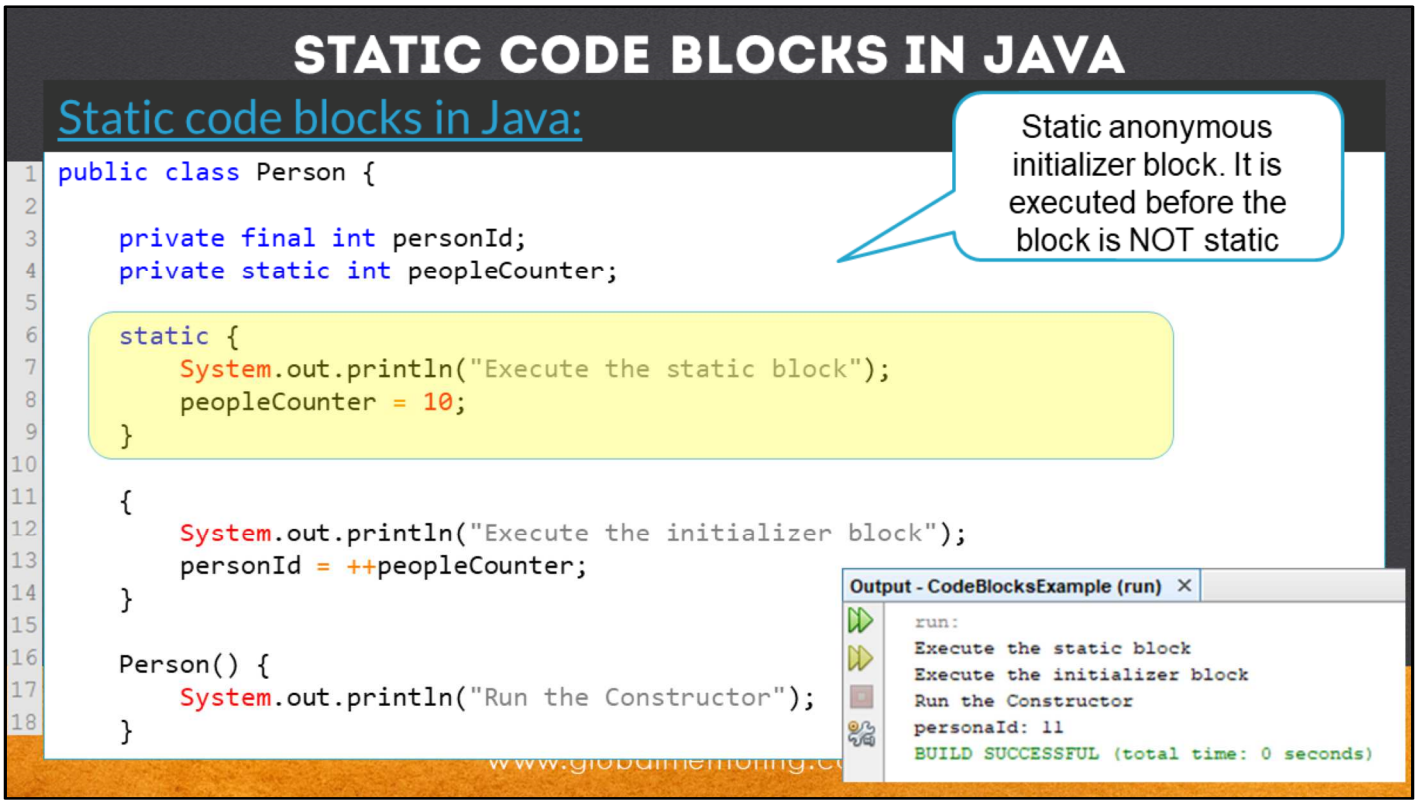
To declare an anonymous type block and initialize instance variables we can see the example in the code shown.

We can see how lines 6 to 9 have been declared an anonymous initialization block, whose objective is that this code be executed before the execution of our constructor, in fact this block of code is copied to each constructor of every object and is executed before the execution of each constructor.

For this reason, in the output of code (output), we can see that the message "Executes initializer block" is seen in console first and then the message "Executes Constructor" is observed. And we also see that the value of the idPersona has been correctly assigned with the help of the static variable CounterPeople.

Recall that the static variables are created first, and then the dynamic variables or class attributes, that is why in line 6 it is possible to use the static variable.

With this we have seen how to use an anonymous code block, which we use to initialize some elements of our classes. This type of code blocks we will see in some cases in the Java classes, and we can already understand how this type of code blocks work.



Now we will see how to initialize the static variables of our code by means of an anonymous and static code block.

Similar to how we saw in the previous sheet, it is possible to use an anonymous code block to perform static variables initialization tasks. For this we declare a block of code with the only difference that we must put the word static and then open and close the curly brackets {}.

Within this code block we can add static code, that is, static variables.

As we observed in the code, we observed that we can change the value of the static variable peopleCounter (line 8) since we will not always have the value we want to assign when we declare our variables, and in this type of code blocks we can add a more complex logic to initialize our static variables in Java.

And here we can observe and conclude the same thing that we studied in the static context lesson. We can see in the output that the static block is executed first, and then the non-static blocks, that is, the initializing block and finally the class constructor.

With this we can understand how the blocks of code are combined to start and build our Java classes, from variables, initialization code blocks and constructors. With this we already have a greater understanding of the construction of our classes.

Let's see an example of the use of code blocks.