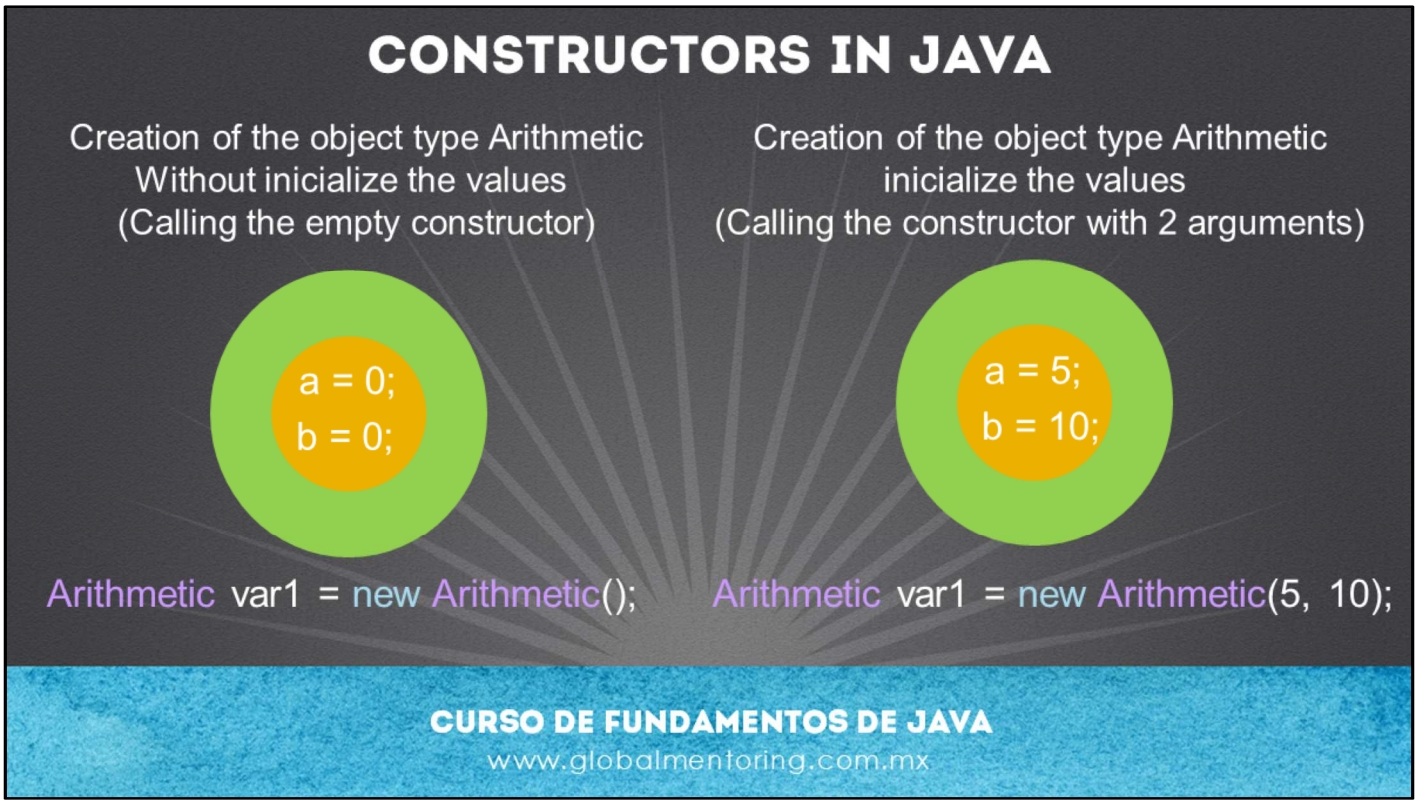
Hello, Ubaldo Acosta greets you again. I hope you're ready to start with this lesson.

We are going to study the subject of methods in Java.

Are you ready? Come on!



It can be very tedious to be initializing the variables of a class each time we create an object, for it Java allows you to add values to the objects from the moment of their creation. This concept is known as Constructor, and basically it is a method but it will allow us to create an object with the values that we provide when creating our object.

The constructors, although very similar to the methods, have certain differences and characteristics, such as:

- These methods can only be executed at the time of creation of an object, it is not possible to use them later.
- They do not return any value.
- The name of the constructor is identical to the name of the class, this is how we can identify if a class has constructors (always respecting the case)
- By default in Java, a constructor without arguments is created, known as an empty constructor. This Constructor adds automatically the Java compiler to our class, however, if we define a constructor other than the empty constructor, that is, with arguments, then Java no longer adds an empty constructor and it is our responsibility to add the empty constructor to our class if necessary.
- The empty constructor is necessary to create an object, remember the general basic syntax to create an object.

  TypeClass object = new TypeClass ();
- If we look at the end of the line of code we are opening and closing parentheses, that is precisely the empty constructor that the compiler added for us as long as we do not add constructors with other arguments. So now we can understand that after the word new really what we are placing is the name of the constructor that we want to call, being able to have arguments or not.

In the figure we can see an example of our Arithmetic class, using an empty constructor and another non-empty constructor to create our objects.

Suppose we have two attributes in our class, of the integer type, called a and b. In the first case we are creating an object by calling the empty constructor, therefore the values of our attributes are initialized with the default values according to the type of each attribute. If it is a primitive type, they are initialized according to their default value, and if they are of type object their value starts is null.

On the other hand we create an Arithmetic object but with initial values from the moment we are creating the respective object. We are sending the values of 5 and 10 respectively, and therefore we can take these values to initialize the attributes of our classes.

CONSTRUCTORS EXAMPLE

General form of a Java Constructor:

```
//General form of a Java constructor
ClassName( arg1, arg2, etc…) {
     //Body of the class constructor
}
```

Example of constructors in Java:

```
// Constructor without arguments
Arithmetic(){
     //Constructor body
}

//Constructor with 2 arguments
Arithmetic( int arg1 , int arg2){
     //Constructor body
}
```

As we have said, a constructor initializes an object at the moment of creation, it is also responsible for reserving the memory for the creation of the object.

To create a constructor in a Java class, the general syntax includes the name of the class, in this way we can identify if a class has a constructor or not. Prior to this, it may or may not have an access modifier such as public or private, these modifiers will be studied in detail later. After the name of the class, similar to a method, it may or may not have arguments.

Remember that Java creates the constructor without arguments automatically, however if we have added a constructor with arguments, then Java will no longer add this constructor for us and it is our responsibility to add it if we are going to need it, in fact for good practice we should always add it that we defined a constructor with arguments.

In Java when we create two or more methods with the same name, this feature is known as overloading of methods (overloading). This means that we can create two or more options when creating an object to create it.

As we observed in the example, we have added two constructors to the Arithmetic class, in the first case it is the empty constructor or without arguments. This constructor can optionally have a code when creating the object, even if it does not receive arguments.

Secondly, we have a constructor with two arguments, and it receives two arguments of type integer, once we receive the arguments, we will normally use them to initialize our attributes of the class. This is what we will put into practice in the following exercise.

Finally, it should be mentioned that a constructor, unlike a method, does not have a return value, therefore we will not find the word return nor a return type in the signature of the constructor. This is because what a constructor returns is the object itself.