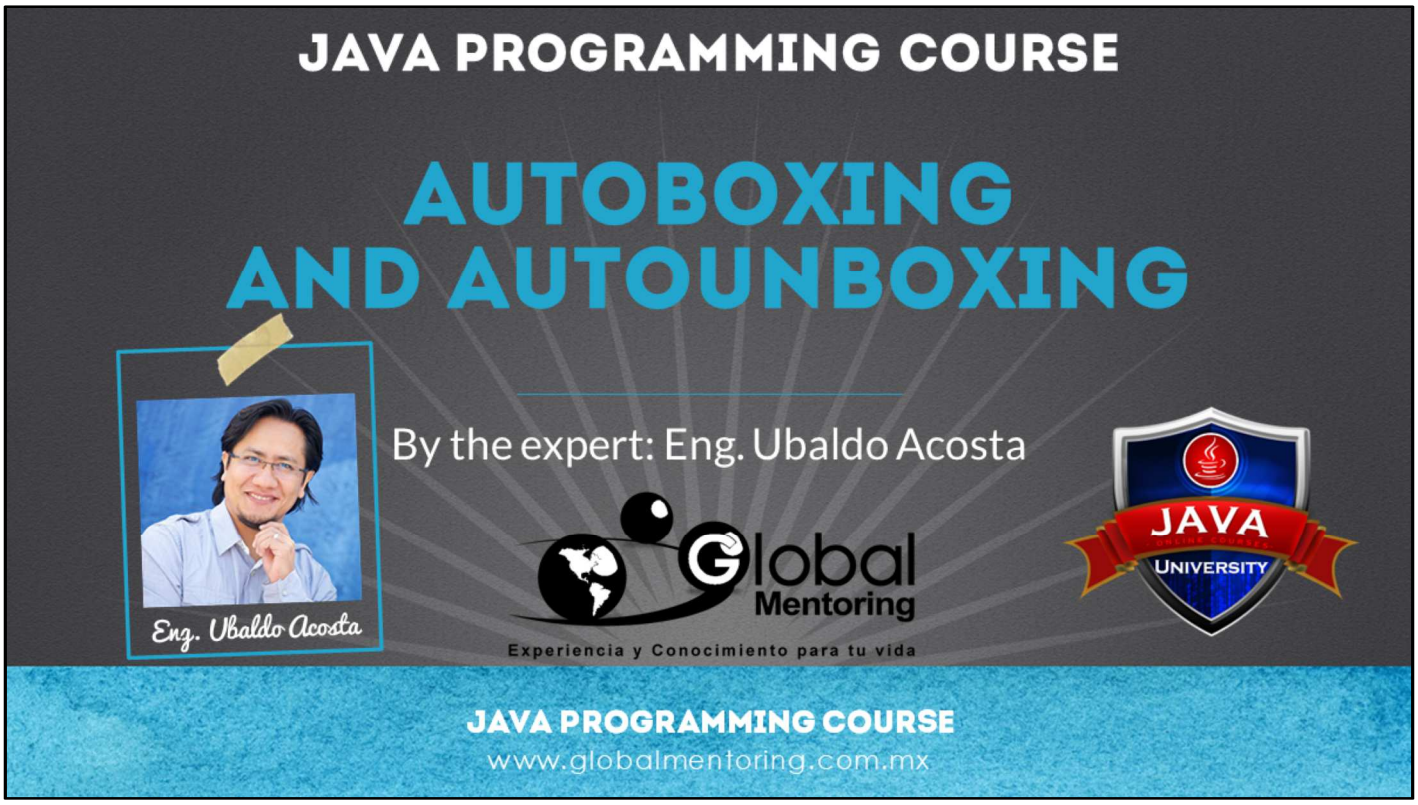
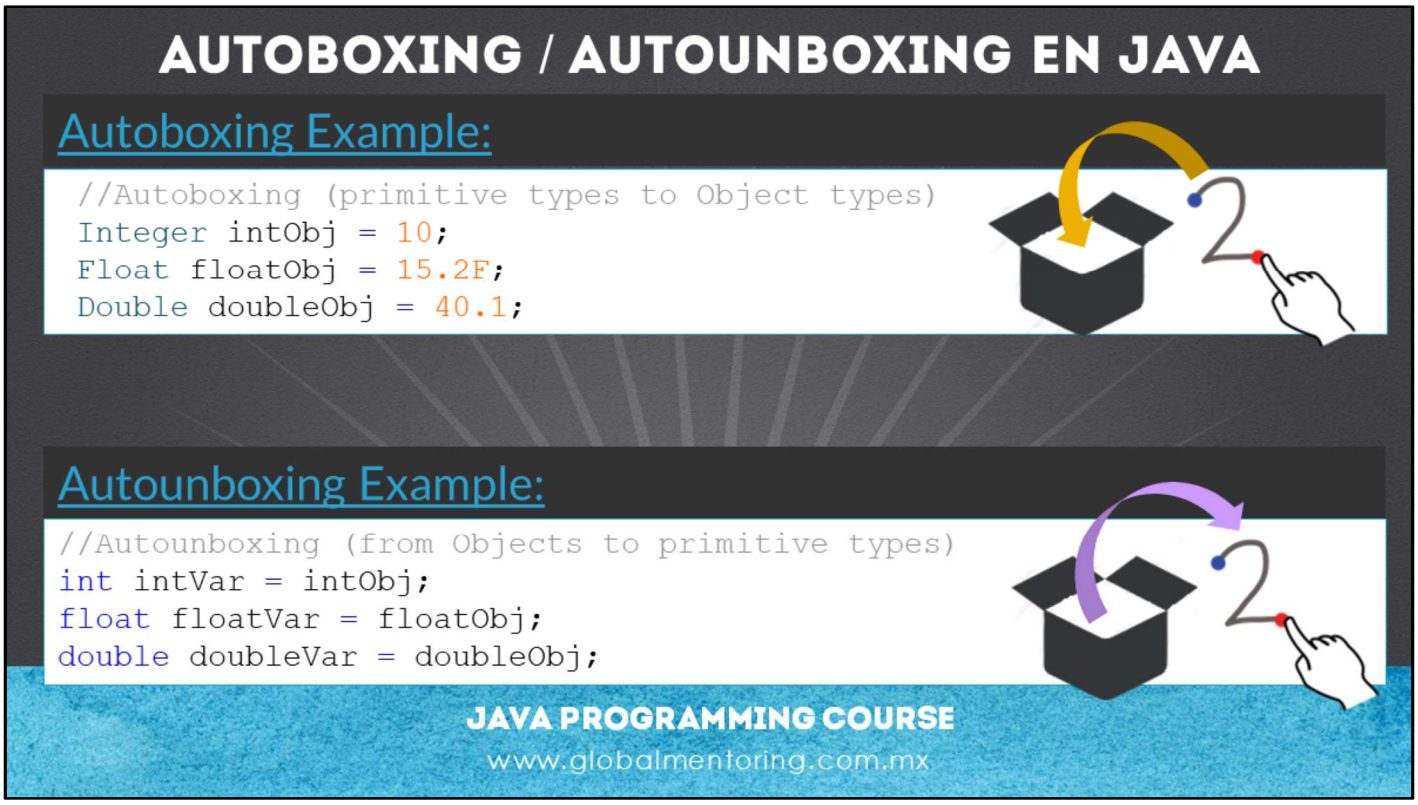


Hello, Ubaldo Acosta greets you again. I hope you're ready to start with this lesson.

We are going to study the topic of Autoboxing and autounboxing in Java.

Are you ready? Come on!

© All rights reserved Global Mentoring

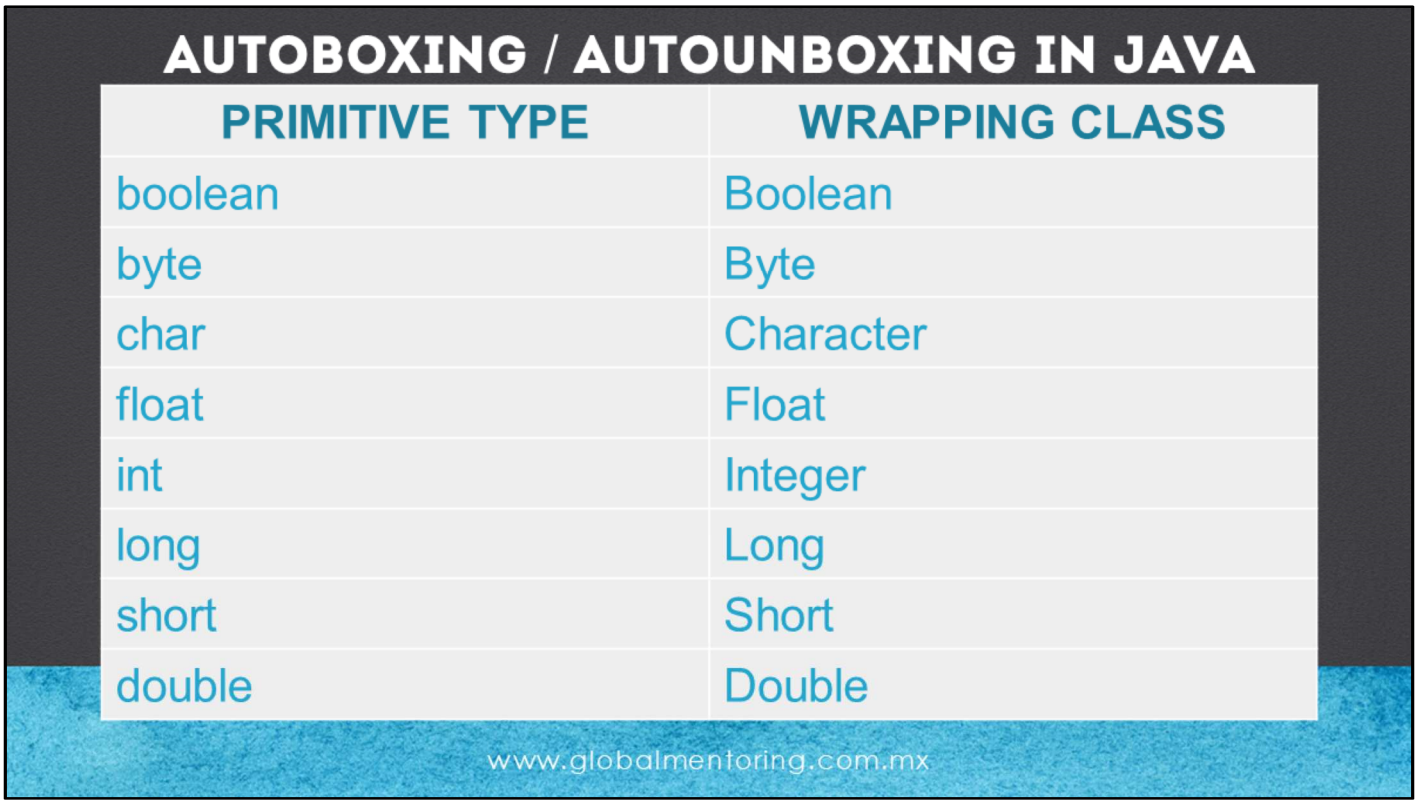


The concept of Autoboxing and autounboxing is simple but powerful.

Autoboxing is the process that the compiler does automatically to convert a primitive type into a Java object in its equivalent Object type. For example, in the code shown we can see that the literal 10, which is a literal of type int, is assigned directly to a class called Integer. This Integer class is known as wrapper and is the one that allows the conversion process automatically between primitive types and objects of type int. There is a type of enveloping class for each of the existing primitive types, which we will see later. The Autoboxing process does not require any kind of conversion between primitive types and objects as long as we use the envelope type equivalent to the primitive type that we are trying to convert correctly.

On the other hand the process of Autounboxing is the inverse process, that is to say, an object of envelope type can be converted automatically to a primitive type without the need to apply any conversion between objects and primitive types, as we can see in the code shown . In this way it is very easy to work with primitive types, but when we need to use the advantages of a Java object we can use the respective envelope class.

© All rights reserved Global Mentoring



In the table shown we can see the primitive type and its equivalent enveloping class. We can see that according to the Java conventions, the envelope class is written with a capital letter, and thus we can quickly identify if we are dealing with a primitive type or with an object.

Several of these wraparound classes descend from a class called Number, and this class in turn is what defines many of the methods that classes that are of a numeric type can use. This class contains the methods to perform the conversion to different types of data as needed, for example, no matter what numerical type is concerned, we can request that you return the corresponding value of primitive type byte, short, int, long, float or double. In this way we can convert the data contained by the enclosing class to any of the types mentioned.

In summary, the concept of autoboxing and unboxing allows to write code simpler and easier to read, without having to make as many conversions between objects and primitive types as it had to be done before version 5 of Java. We are going to develop an exercise to apply this concept.

© All rights reserved Global Mentoring

© All rights reserved Global Mentoring